Matching of image features and vector objects to automatically correct spatial misalignment between image and vector data sets

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Science

At the

University of Canterbury

Daniel Gerard O'Donohue

26-02-2010

Abstract

Direct georeferencing of aerial imagery has the potential to meet escalating demand for image data sets of increasingly higher temporal and spatial resolution. However, variability in terms of spatial accuracy within the resulting images may severely limit the use of this technology with regard to operations involving other data sets. Spatial misalignment between data sets can be corrected manually; however, an automated solution is preferable given the volume of data involved.

This research has developed and tested an automated custom solution to the spatial misalignment between directly georeference aerial thermal imagery and vector data representing building outlines. The procedure uses geometric matches between image features and vector objects to relate pixel locations to geographic coordinates. The results suggest that the concept is valid and capable of significantly improving the spatial accuracy of directly georeferencing aerial imagery.

Acknowledgments

I would like to thank my two supervisors, Dr. Femke Reitsma and Dr David Park, your support and insightful comments have been a huge help.

Thanks to all the folk at the Geospatial Research Centre, my involvement with you represents the most significant part of my university career. In particular I would like to thank my mentor and friend, Phil Bartie, who, without knowing it has been the testing ground of many ideas and the source of much inspiration.

Lastly I would like to thank my beautiful wife. Without her patience and support I would never have made it.

Table of Contents

1. Introduction

The aim of this research is to design, implement and analysis a custom automated solution to the spatial misalignment between directly georeferenced aerial thermal imagery and vector representations of building outlines. The methodology attempts to match the geometries of images features with building outlines within a distance tolerance. Successful matches provide Ground Control Points for the process of georeferencing. Although the methodology implemented during this study focuses exclusively on the misalignment between the aforementioned data sets, the underlying concept is adaptable and could be applied to other image and vector data sets.

Images provide a means of capturing an instantaneous representation of a given scene. This ability is not limited to representations consisting solely of values derived from the visible spectrum. Current sensor technology is able to record and represent scenes based on levels of reflected and emitted radiation in a variety of different sectors within the electromagnetic spectrum. The advent of digital imagery has meant that the information stored within images is exploitable by ever increasing computing power and software designed to perform operations that manipulate image data.

Geographical Information Systems (GIS) data is available in raster and vector formats. Operations within a GIS often involve the use of both types of data. Operations of this nature involve data fusion, although, this term can also be applied to the use of different data sets of the same format. Fusing data of the same format is a well established practice (Zitova and Flusser, 2003). For the purposes of this research, image and vector data fusion will be considered to be the use of image and vector data in such a way that the derived interpretations and representations are of a higher quality than those resulting from the use of a single image or vector data set (Hahn and Baltsavias, 1998). However, the fusion of image and vector data is complicated by the two differing methods of representing objects and their attributes. Vector data combines geometric information with explicitly stated attribute data. Where images provide a continuous representation of a geographic area, vector data represents objects as discrete entities. The successful fusion of raster and vector data has the potential to provide significant added value compared to the possible output resulting from the use of a single image or vector data set.

An example of this type of data fusion is the use of vector data to select image classification training sites within georeferenced imagery (Mena and Malpica, 2005). Training sites refer to areas containing a known surface type. During image classification the pixel values in these areas are then used to select areas with similar pixel values and, in theory, similar surface covers. Using vector data in this way means that the usual validation of training areas could be avoided by using vector objects to confirm the accuracy of the classification results. The use of vector data and image data has also successfully been applied as a means of updating existing spatial databases. This process involves using vector objects to identify image features, once identified the attributes of vector objects can be updated with information derived from the imagery (Ehlers, et al. 2003; Gamba, et al. 2005; Weis, et al.). For example, aerial and satellite images are often used to investigate the physical prosperities of surfaces based on reflected or emitted radiation. Assuming that an existing vector object represents the extent of a forest and spatially intersects with an aerial or satellite image, the image area within this intersection could be classified as forest based on the attribute data associated with the vector object. Furthermore, the vector object could receive additional attribute values based on the reflected or emitted radiation values sorted within the image. The underlying principal is that vector data represents specific information about the geometry, topology and attributes of a given object. This information is then able to be used to validate the existence of an object at a given location, which is either confirmed, denied or flagged for further investigation dependant on the results of the image classification routines used. An example of this is the use of vector data as a means of classifying road segments occluded by trees or buildings (Bonnefon, et al. 2002). Obviously, the process of detecting new objects within imagery is less able to make use of existing vector data, assuming that new objects are objects that do not already exist as vector objects (Baltsavias, 2004). However, it does seem reasonable to assume that confirmation provided by vector data that an object does not exist is also of use when seeking to establish that the detected object is in fact new.

The advantages of such application are immediately obvious, vector objects are used during the process of image classification and the information derived from images can be used to update the attribute information associated with vector geometry. This process is simple when considering relating a well defined individual feature in a single image to an equally well defined object represented in a vector database. Given this situation, a skilled GIS user is easily able to manually relate image features to vector objects or a simple spatial intersection operation may achieve the same result. The relationship may be based on matching geometry, spatial intersection, colour recognition or as a result of a combination these features. However, in a situation with thousands of images, each with multiple features, manual processing is impractical. This situation is further complicated if spatial alignment between image features and vector objects differs on an irregular basis. This is the likely outcome of calculating position and orientation of aerial images using the process of direct georeferencing.

There can be little doubt that the collection and consumption of spatial data is increasing along with the demand for increasingly higher temporal and spatial resolutions. Imaging systems combined with photogrammetry processes have proven their ability to produce vast quantities of detailed geographic data (Cleve, et al. 2008). An example of this is the process of direct georeferencing of aerial imagery. Direct georeferencing calculates the location of the image based on the position and attitude of the sensor at the time the image was captured in conjunction with knowledge of the internal and external distortions within the sensor and produced by the topography (Skaloud and Legat, 2008). The Global Positioning System (GPS) produces precise positions that are subject to errors arising from loss of satellite lock and resolution of phase ambiguities. Information from Inertial Navigation Systems (INS) can be used to correct these errors while the GPS data continuously calibrates the INS. This provides a solution to positioning and orientation of the sensor at the time an image is captured. By accounting for the internal and external distortions, an image can be geographically located without the need to manually specify Ground Control Points (GCPs) (Jacobsen, 2002). The application of direct georeferencing enables high resolution aerial imagery to be captured over large areas with limited recognisable terrain features. Achieving this same task manually would require detailed knowledge of the area in the form of existing spatial data to which image features could be related and the patience and time of a skilled GIS operator. Despite advantages in terms of the amount of resource and time required, the spatial accuracy of the georeferenced imagery produced by direct georeferencing can vary dramatically from image to image due to slight irregularities in the calculated position and orientation of the sensor.

The previous examples of image and vector data fusion demonstrate the potential of combining images and vector data for the purposes of validating assumptions about image features and enriching vector object representations of real world features. However, the previously mentioned applications rely heavily on the spatial alignment of image and vector objects. Direct georeferencing has the potential to meet the demand for increasingly higher temporal and spatial resolution, yet spatial variability produced by this method may limit the use of this technology in terms of fusion with vector data. One approach to solving the problem of spatial variability is the manual manipulation of individual images. However, given the quantities of data capable of being produced by direct georeferencing an automated solution is desirable.

Spatial alignment is a key factor in the process of fusing image and vector data. The goal of this research is to provide a means of spatially aligning inaccurately georeferenced aerial imagery with existing vector data. Alignment will be based on the geometric matching of image and vector objects. The first step in this process involves the conversion of image features to a comparable abstraction level. Geometries will then be compared and topological attributes will be used to isolate a single matching pair of vector objects. Successful matches will identify common locations within image and vector data sets and thus allow the location of the image to be updated to align with that of the vector data. This methodology will be applied specifically to a data set consisting of directly georeferenced thermal images and vector representations of building outlines. This process has the potential to provide a solution for the misalignment between large image data sets and existing vector data, thus facilitating the process of image and vector data fusion.

Section 2 provides an overview of manual and automated methods of image registration as this is fundamental to the process of image and vector data fusion. Section 3 focuses exclusively on the use of matching image features to vector objects as means of georeferencing images. Section 4 describes the methodology implemented during this research and section 5 presents the results and discussion. Conclusions, research limitations and suggestions for future research are offered in Section 6.

2. Image Registration

One of the prerequisites for image and vector data fusion is a common registration and as such is the focus of the following discussion (Baltsavias and Hahn, 2000).

Registration is the process of assigning coordinate values to pixel locations or vector objects. It is distinguishable from co-registration in that it refers to a global reference frame such as a geographic coordinate system (Muller, et al. 2007). Co-registration refers to objects that are comparable with respect to a given aspect. Although objects can be co-registered in relation to aspects such as time and spectral properties, the ability to relate objects in terms of spatial context is central to the process of fusing image and vector data (Baltsavias and Hahn, 2000). In practical terms, image and vector data fusion is dependent on the registration of datasets as opposed to co-registration. The accuracy of registration is also important, poorly aligned image features and vector objects complicate the task of fusion. The retrospective improvement of image georeferencing is possible via manual editing and automated procedures.

2.1 Manual Registration

Images may be manually registered or georeferenced. The process usually involves identifying an object in both the image to be georeferenced and in another data source that is registered to a coordinate system. This is often within spatial editing software suites that allow the user to visually allocate geographic coordinates to pixel locations based on matches between objects. Assuming that the other data source is spatially accurate the assignment of geographic coordinates to pixel locations should result in acutely georeferenced image. However, it is important to realise that both the number of matches between objects, the distribution of the matches within the image and the resulting transformation used to define the relationship between pixel and geographic coordinates, ultimately define the spatial accuracy. In order for this process to be successful, the operator must be able to accurately match image features with objects in the other data source. Manually georeferencing is a time consuming task and requires a skilled operator to physically assign geographic coordinates to features within the image based on the recognition of the same feature within another data set. Considering the volume of georeferenced imagery that direct georeferencing of aerial imagery is capable of creating, manual manipulation and correction is simply not practical.

2.2 Automated Registration

Current methods of automatically correcting the georeferencing of imagery include image to image and image to map registration techniques as well as bundle adjustment algorithms and matching image features to vector objects. Image to image and image to map registration involve matching features in multiple images of the same scene possibly taken at different times, from different sensor locations and potentially also different sensors (Zitova and Flusser, 2003). Registration produces a mosaic of the scene. Mosaics produced by image to image registration define pixel locations in terms of pixel coordinates, whereas an image to map registration locates pixels and therefore imaged features in terms of geographic coordinates.

The basic principle behind this approach is to locate an area of one image (slave image) in another image (base image) (Scambos, et al. 1992; Snow National Center Ice Data 2009). Successful location of several regions of the slave image in the base image result in points of best fit. These points of best fit are then treated as GCPs and used to link the slave image to the base image. An obvious problem with this technique is that overlap between the two images controls the distribution of possible GCPs. Clearly, GCPs can only be located within the overlapping areas of both images, thus, unless significant overlap exists, GCPs will be poorly distributed throughout the slave image. The effect of this is that the geographic accuracy within the slave image may decrease dramatically with distance from the overlapping area due to transformation properties.

As previously stated the process of direct georeferencing calculates the location of the image based on the position and attitude of the sensor at the time the image was captured in conjunction with knowledge of the internal and external distortions within the sensor and produced by the topography. In practice the direct georeferencing of aerial imagery for use within a GIS requires highly accurate GPS and INS instruments in combination with custom software. By accounting for the internal and external distortions, GCPs can be automatically located within the image. Bundle adjustment if often applied to the output of the direct georeferencing process in an effort to improve the spatial accuracy (Mouragnon, et al. 2009). Bundle adjustment takes a set of images captured from various viewpoints with a number of known x,y and z locations in each image and refines the estimation of the coordinates. This is achieved by minimising the reprojection error between observed and predicted points within each image, using a nonlinear least squares solution (Goncalves and Aruajo, 2009). The refined x,y,z locations define transformations that mathematically describe the relationship between pixel coordinates and their corresponding geographic location. Bundle adjustment is widely applied throughout the photogrammetry community (Yuan, et al. In Press). Although bundle adjustment is well established, the output is still reliant on accuracy of the initial known 3 dimensional points in each image. Bundle adjustment also assumes the availability of the highly accurate location and attitude data for the sensor at the precise time that each image was taken.

The utility of matching features in aerial imagery to vector representations of the same real world objects is not a new realisation. Successful matches have been used to determine common points within both data sets (Morgado and Dowman, 1997). On the basis of these common points or GCPs, the image was registered to the vector data set and thus a common coordinate system. Image and vector data representations of objects are fundamentally different and as such conversion from one representation to the other is required in order for matches to be possible. The implementation of shape matching in the field of GIS has seen the matching process aided by the use of geographic data sets such as elevation models and aspatial data (Xiong, 2000; Zhang, 2004). Matches between objects and groups of objects have also been sought at varying spatial resolutions (Benz, et al. 2004). GIS shape matching implementations have successfully related patterns comprised of individual points from various features in one data set to corresponding patterns in another data set (Schuurman, et al. 2006).

This research focuses on the use of matching image features to vector representations of real world objects in order to accurately georeference imagery. Matches are sought between image representations of buildings and vector building outlines, whereas, many previous methodologies have often focused on the use of road networks. This may reflect the ease with which roads can be segmented from images or the availability to data. During the implementation of this research, matches are considered successful if the geometries match within a distance tolerance. The spatial relationship between individual matching pairs and all matching pairs for a given image is considered in order to filter mismatched objects. Thus an effort is made to consider patterns as well as individual matches between objects. Similar to bundle adjustment, the methodology implemented during this research assumes that images are already registered to a geographic coordinate system, however, bundle adjustment requires detailed knowledge about the location and attitude of the sensor at the time of imaging combined with the existence of overlapping imagery. This research considers images on an individual bases and no information regarding the position or attitude of the sensor is required.

The following section provides a detailed discussion of the fundamental sub-processes involved in identifying and extracting images features from a GIS prospective. This is followed by an overview of methods used to calculate matches between geometries.

3. Matching Image Features To Vector Objects As A Means Of Georeferencing Images

In order to match image features to vector data, image features need to be converted to the same abstraction level as the vector data. The complexity of reality means that real world objects and their attributes need to be simplified in order to be represented within a computer, this is termed abstraction (Longley, et al. 2005). Abstraction level therefore denotes the level to which a real world object is simplified. In order for image data to be matched with vector data, image features need to be represented on the same abstraction level as vector objects. Image data is stored as pixel values, and unless objects within the image are labelled the meaning that can be assigned to each pixel value or group of values is at the user's discretion. Therefore objects within images can be seen as implicit rather than explicit. Although objects within images may be easily recognisable to human interpreters, it is not often explicitly stated what each feature represents, the exact geometry or what attributes it might have. Feature type, geometry and attributes are implied given the interpreter's knowledge and experience. Vector data can contain explicit information about the represented object. The objects geometry is explicit in that it consists of a single point or multiple linked points that define a discrete entity. The location of each point is explicitly defined and vector objects are also able to contain specific topological and attribute information. Without the conversion from image data structures to vector data formats, the feature matching of these two data sources would be problematic. Image classification and manual digitising routines provide conversion from image data to vector compatible abstractions.

3.1 Image Classification

Although it is possible to convert vector representations of objects to raster representations of the same objects, it is unlikely that the original image can be reconstructed from vector data. Thus comparisons are usually made between vector objects. Comparing image and vector data cannot be done at pixel level. High resolution imagery effectively means that single pixels lack the aggregating effect of low resolution images, therefore reducing the likelihood that objects are adequately represented by individual pixels. Consequently, comparison between image and vector data types can only be accomplished after pixels have been aggregated into meaningful objects (Weis, et al. 2005). This aggregation is often termed image segmentation or image classification (Liu, et al. 2007). Image classification refers to the grouping of pixels into classes based on spectral and or geometric properties (Costa and Cesar, 2001; Weis, et al. 2005). Manual image classification may be purely based on the ability of a human operator to recognise image features and categorise them into predefined object classes, whereas common automated sub-routines within the image classification process utilised actual pixel values. These sub-routines are algorithms which automatically carry out an operation on a given image; however, parameters are often user defined and outputs manually assessed.

Supervised and unsupervised classification routines are examples of automated sub-routines that use either pixel-based or object-based techniques. Supervised classification differs from unsupervised classification in that classification categories are user defined based on prior knowledge and then applied to the whole scene, whereas unsupervised classification assigns pixels into a given number of categories based solely on statistical similarities (Richards, 1993). Unsupervised classification is often performed using clustering methods. Common unsupervised classification routines such as Iterative Self-Organising Data Analysis Technique (ISO - DATA), assign pixels into groups based on distance from the class means in data space.

3.1.1 Pixel-based

Pixel-based classification routines group pixels into classes based solely on their spectral properties. High-resolution imagery reduces the aggregating effect of large pixels, and therefore increases the spatial variation within imaged objects (Durieux, et al. 2008). Although this may lead to more detailed spectral signatures to base classifications on, it will also increase the variation of within uniform surfaces (Schiewe 2005; Cleve, et al. 2008). The sole use of pixel-based classification techniques may result in logically uniform surfaces being classified as several different objects due to high levels of spectral variation within the pixel values (Ehlers, et al. 2003).

3.1.2 Object-based

Object-based classifiers overcome many of the issues surrounding pixel-based classification routines by first segmenting the image into groups of neighbouring pixels and then classifying these objects into categories. By grouping pixels into objects first, the classification algorithm is able to consider spectral, topological and contextual characteristics of each object when determining which class each object should be assigned to (Cleve, et al. 2008).

The process of image classification is often subject to validation procedures. Traditionally, classifications produced by the process of manually editing an image have been accepted as being correct. However, given the complexity of classification there is a need to assess the reliability of the results (Congalton, 1991). This may involve the collection of ground truth data or if data deemed to be sufficiently accurate is available this could be used to aid the classification process. In either case the classification results are usually compared to another data source in order to quantify the accuracy of the classified image. A common method used to characterise site specific accuracy is an error matrix (Congalton, 1991). Error matrices provide an assessment of the alignment between the reference data and classification data, but are based on the assumption that each pixel can only belong to one class (Binaghi, et al. 1999). Another approach is to use fuzzy set theory, which allows pixels to be assigned to multiple classes. This means that accuracy can be quantified by means of assessing the level to which each pixel belongs to a given class. This type of soft classification has been successfully used to classify mixed pixels (Binaghi, et al. 1999; Xu, et al. 2005).

Conversion from image to vector representations may also result in the vector object appearing stepped. This is a legacy of representing data as a series of tessellating rectangles or pixels and complicates the geometric comparison process. This could dramatically alter the area and dimensions of the objects making it difficult to relate existing vector representations of the same object that have not been converted from image features. For this reason, and to avoid computational overhead during processing, objects are often simplified.

3.2 Object Simplification

Object simplification aims to reduce the number of points that constitute an objects boundary while retaining the main features of the objects original geometry. Points that best represent the original geometry might be selected based of maximal curvature, distance from the centroid or other criteria suitable for the particular shapes involved (Zhang, et al. 2003; Super, 2004). The Douglas-Peucker algorithm is a commonly applied to simplify geometries within the field of GIS and is an example of simplification on the basis of curvature (Erbisch, 2002). In theory, simplified objects contain less information and as such are less resource hungry in terms of computer power required during display and processing. The generalization of objects also plays an important role in cartography by enabling users of cartographic outputs to gain a quick overview by reducing the level of detail displayed (Cheung and Shi 2006).

Having derived meaningful objects from image data via image classification and object simplification it is then possible to compare the derived data with existing data. While manual methods of locating, creating and evaluating vector comparable representations from image objects are practical for several images containing a limited number of easily identifiable features, the task soon becomes impractical when dealing with larger numbers of images. The ability to automatically identify and relate image features to vector objects would greatly increase the usability of large image data sets that are spatially misaligned within corresponding vector data.

3.3 Approaches To Shape Matching

Shape matching is a central problem in areas such as computer vision, pattern recognition and robotics. Shape matching is often applied during industrial inspection, content based image retrieval and fingerprint inspection. There is a large base of literature in this area focusing on the representation of shape, criteria for matching and algorithms that have been applied to this problem (Loncaric, 1998; Mustafa, 2002). Matching criteria are usually selected on the grounds that they are scale, rotation and translation invariant (TU, et al. 2008). In practise is it likely that robust matches are based on shapes having multiple similarities in terms of the chosen matching criteria. Given the volume of literature in this area the following discussion seeks only to provide an overview of current methods of defining matching criteria and representing shape and should not be viewed as a definitive resource.

Global shape matching methods focus on the use of the complete object area or contour and as such assume that the complete contour or area is able to be extracted from the image. Having extracted the contour of an object, global shape matching methods base matches on criteria such as area, circularity, eccentricity, compactness, major axis orientation, Euler number, concavity tree and shape numbers (Bernier and Landry, 2003). Using Curvature Scale Space involves contour smoothing by way of Gaussian filter of increasing standard deviation (Zhang, 2003). After smoothing, the points where the curvature of the contour changes from positive to negative are recovered and mapped to a graph where the horizontal axis represents the arc on the original contour and the vertical axis represents the standard deviation of the Gaussian filter. The resulting graph uniquely represents the original contour and as such comparisons between graphs provide a method of matching shapes. Point based shape matching methods often use local geometric features, such as curvature, angle and arc length to identify mathematical landmarks on the shape boundary (Rueda, et al. 2010). By locating these landmarks in another shape a measure of similarity between the two shapes can be gained. Shape can also be represented and compared by means of chain code. Chain code represents rasterised shapes as a string of directional values based on 4-8 possible directions (Kato, et al. 2007). Shapes can also be expressed using Fourier descriptors and comparisons based on these descriptors provides a means of matching geometries (Zhang and Lu, 2003).

The shape matching procedure implemented during this research uses a system of positive and negative buffering operations in combination with simple topological tests to establish if geometries matched. This method was chosen for its conceptual simplicity and its ability to consider scale and rotation. Furthermore, the operations used are fundamental concepts in the field of GIS. The following section describes the methodology used during this research in detail.

4. Methodology

4.1 Data and Processing

The data available for this research consisted of 394 directly georeferenced thermal images in geotiff format and building outlines in shapefile format. The thermal images were captured between 2 and 3pm over the township of Rangiora, Canterbury, New Zealand on 06/02/2009. Thermal images were captured using the Flir PathfindIR thermal camera that recorded thermal emission within the spectral range of 8 to 14 micrometers. Building outlines were created via heads up digitising of manually georeferenced colour aerial images with a spatial resolution of approximately 50 cm (*Figure 1*). It is important to note that not all buildings visible within the images were digitised. Buildings were not digitised if the complete outline of the building could not be adequately identified. This lack of contrast between buildings and the surrounding environment was often the result of overhanging vegetation and similarity of colour between roofs and the surrounding landscape.

Figure 1 displays the colour aerial images used for the manual digitising of building outlines. Successfully digitised buildings are shown with red outlines.

The following methodology describes the custom algorithms written in C Sharp and using the NetTopologySuite and C Sharp bindings for the Geospatial Data Abstraction Library (GDAL). The goal of this methodology is to provide a means of identifying and extracting buildings from directly georeferenced thermal aerial images, matching extracted buildings against vector representations of the same buildings and to use successful matches as a basis for spatially aligning the two data sets. The methodology is logically separated into the following four major processes *(Figure 2);*

- Image Thresholding and Contouring
- Selection of Polygons for Comparison
- Shape Matching
- Correcting the Original Image

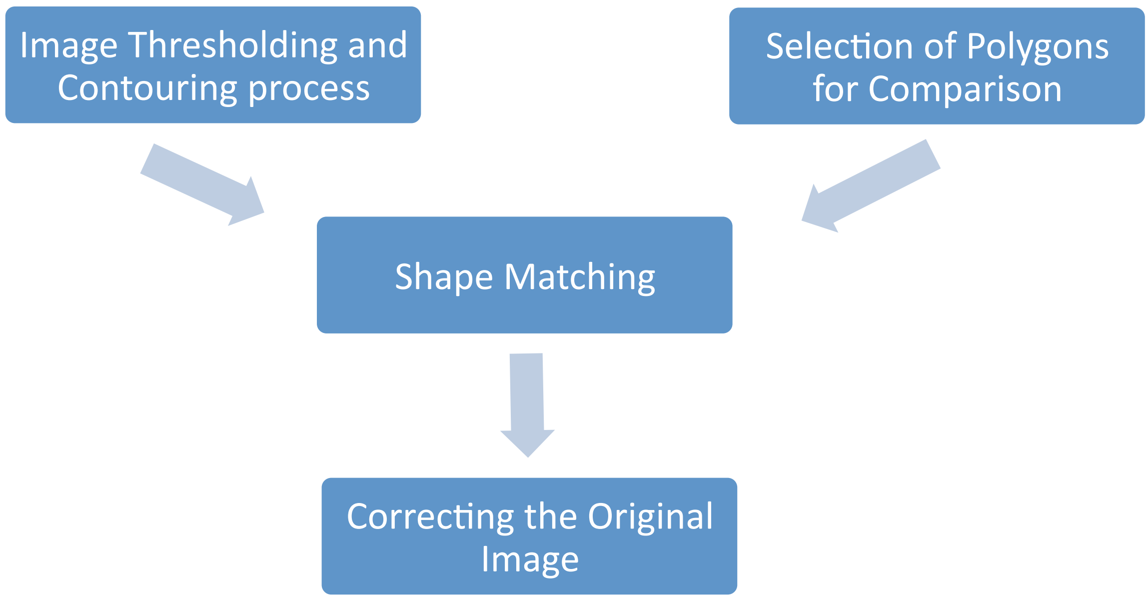


Figure 2 provides an overview of the four major processes involved in this methodology and demonstrates the order in which processes are implemented.

Each of the processes depicted in Figure 2 can be separated into smaller individual sub-processes. These major processes are explained by examination of the aforementioned sub-processes.

4.2 Image Thresholding and Polygon Extraction

Image thresholding and polygon extraction is comprised of the following sub-processes *(Figure 3);*

- Iterative threholding
- Edge detection
- Polygon filtering
- Polygon extraction
- Coordinate conversion

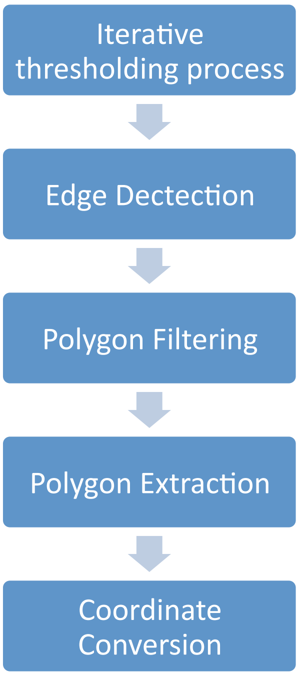


Figure 3 depicts an overview of the process of identification and extraction polygons from thermal imagery

These sub-processes are designed to identify and extract vector representations of buildings from the thermal images and convert the coordinates of extracted buildings from pixel space to geographic space. A detailed description of each sub-process follows.

4.2.1 Iterative Thresholding Process

Pixel values in thermal images represent radiance values in the thermal spectrum. Although the images considered during this research consist of values for red, green and blue, each pixel has the same value for each of these colours. This is a legacy of the pre-processing techniques that create single images from the video output of the thermal camera. As such, each image is a 24bit greyscale representation of the relative thermal radiance of objects within a scene. Therefore, the classification of pixels based on detailed spectral signatures is problematic, given that pixel values can only range from 0 to 255 and represent a single section within the entire electromagnetic spectrum. Classification of pixels as buildings is further complicated by the fact that it can not be assumed that all buildings have similar radiance values. Recorded building radiance values will vary depending on material type, construction style, aspect and internal/external heating at the time of imaging (Oke, 1993). However, as this particular dataset was captured on a relatively warm day during daylight hours, it is reasonable to assume that buildings are likely to be warmer and radiate more thermal energy than the natural environment (King and Davis, 2007). In practical terms, this assumption means that buildings are most likely to have relatively high pixel values. The fact that PathfindIR thermal camera automatically recalibrates to ensure a high level of contrast within the images reinforces this assumption. Although this means little in terms of using specific pixel values to accurately classify buildings, it does mean that low pixel values can be excluded. Given these limitations, classification based on geometric properties as opposed to spectral characteristics has a greater likelihood of providing accurate results.

In order to achieve this it was first necessary to extract features from the imagery that may represent buildings. Feature extraction was based on an iterative thresholding method that produced image contours. Applying a threshold to an image results in a binary or black and white image. Threshold application consists of comparing every pixel value in an image to a given threshold, all pixel values equal to or greater than the threshold are made white and all pixel values below the threshold are made black *(Figure 4)*.

In order select appropriate threshold values for each image the mean pixel value was calculated along with the standard deviation. The mean value represented the start of a continuum, with the end being two standard deviations from the mean. Each value in this continuum was set as a threshold.

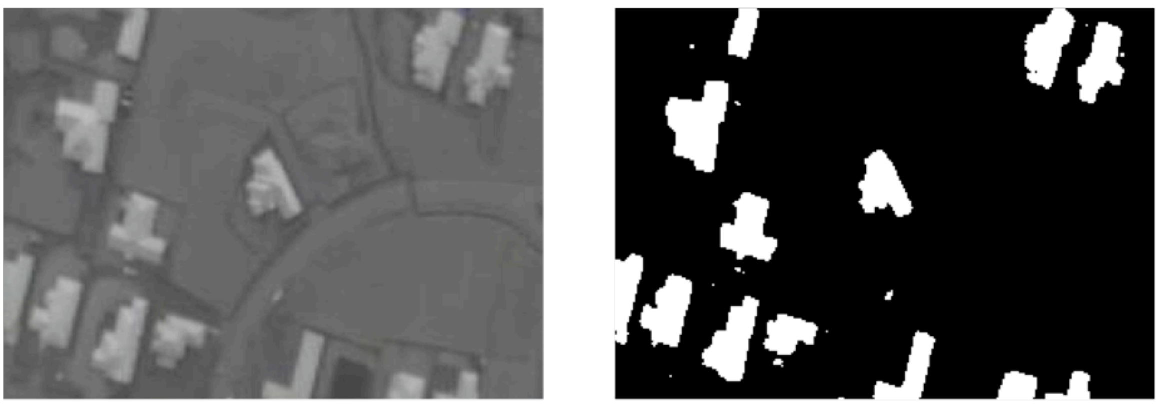


Figure 4 shows the original image on the left and the result of the thresholding process (binary image) on the right.

As previously stated, it could not be assumed that all buildings have similar pixel values. Had this been the case it would have been possible to manually identify a building and select all other buildings using a range of pixel values from the first. However, it is reasonable to assume that all pixels representing an individual building's roof have similar pixel values. The binary image produced via the thresholding process affectively segments the image into groups of pixels. These pixel groups represent areas of similar pixel values and therefore potentially the roofs of buildings *(Figure 4)*.

This method was preferred over edge detection algorithms for image segmentation. Edge detection algorithms highlight the level of contrast between individual pixels and neighbouring pixels. Given the previous assumption that buildings will have higher pixel values than the natural surroundings, edge detection appears to offer a reasonable solution to the problem of extracting building geometries from images. Another advantage of this approach is that edge detection is achievable in a single pass i.e one operation that looks at each pixel and its neighbourhood, as opposed to the previously described iterative thresholding method. While edge detection appears suitable, it results in a range of pixel values as opposed to binary images. This range of values represents the amount of contrast between a pixel and its neighbourhood *(Figure 5)*. The variable amount of contrast is demonstrated in the middle image of Figure 5. As seen in Figure 5, edge detection does highlight the outline of each building, however, the outlines of other contrasting areas are also highlighted. It may be a natural to assume that if only the brightest of all the bight pixels are selected than the outlines of the buildings could be extracted. The right image in Figure 5 demonstrates the results of only considering pixel values above a given threshold. While, only the outlines of the buildings remain visible, the outlines are no longer closed polygons. Both the decision as to which pixel value represents the edge of a building as opposed to the contrast between other surfaces and how to close the resulting open polygons is problematic. Assuming that a range of possible pixel values was selected, it is highly unlikely that this would result in closed polygons and the decision as to how to close them would almost certainly result in a geometrically incorrect representation of the original shape. This in turn would greatly reduce the chances of achieving accurate matches between image derived polygons and database polygons.

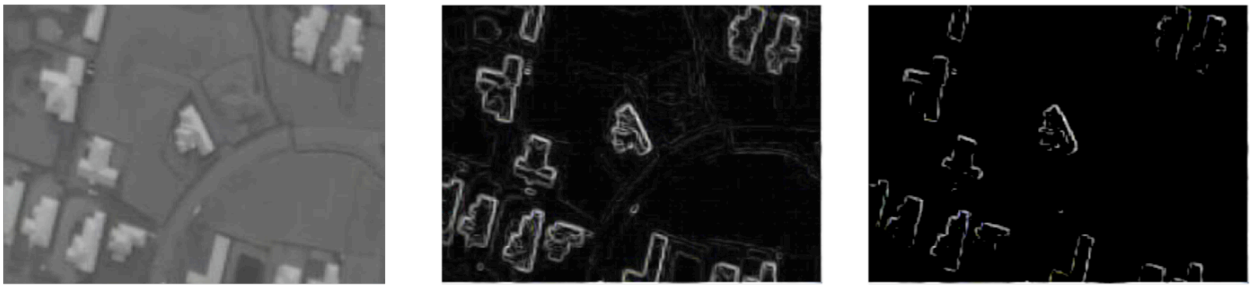


Figure 5 shows the results of applying an edge detection algorithm to a thermal image before the thresholding process has taken place. The original thermal image (left), the result of the edge detection algorithm (middle) and applying a threshold to the image produced using edge detection (right). The image to the right demonstrates the difficulties involved creating closed polygons from the results of edge detection.

The distinction between edge detection as a means of image segmentation and edge detection as a method of isolating the boundary of segmented objects is important. The previous section focused on the use of thresholds to identify buildings within images and justified the application of this process as opposed to use of edge detection for the purpose of image segmentation. The following section advocates edge detection as a means of isolating the boundaries of previously identified objects in binary images.

4.2.2 Edge Detection

The sharp contrast in the binary images produced by the thresholding process meant that edge detection could now be applied to create contours representing the boundary of each feature *(Figure 6)*. The matching process is based on geometry and as such only polygons representing the boundary of grouped pixels were of interest.

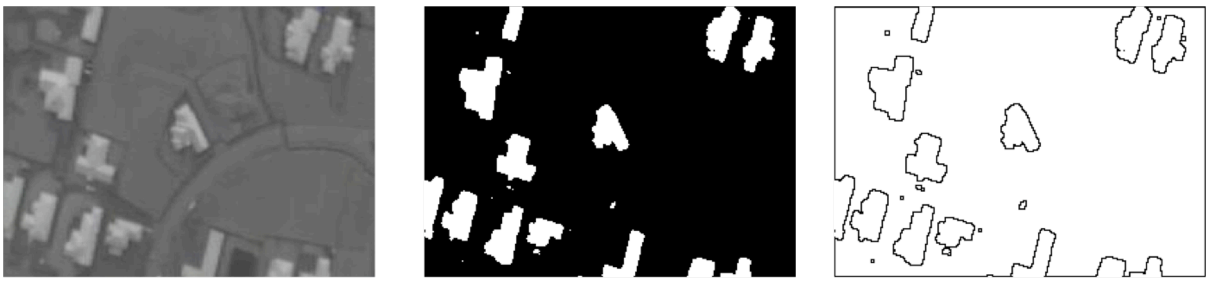


Figure 6 shows the original image (left), the result of the thresholding process (middle) and the result of edge detection on the binary image resulting from the thresholding process (right).

As no out of the box edge detection function was available within the standard C Sharp image processing library, a custom algorithm was designed. Given that edge detection was to be carried out on binary images, this process simply needed to locate all white pixels adjacent to one or multiple black pixels *(Figure 6)*. All white pixels were compared to their adjacent pixels; if any adjacent pixels were black the current white pixel location was considered an edge pixel *(Figure 6)*. This routine ignored pixel locations on the edge on the image, and in so doing provided a simple method of detecting polygons that intersect the image edge and therefore are unlikely to represent the complete extent of the real world object.

4.2.3 Polygon Filtering

With regard to the outcome of edge detection, it is important to note that polygons which intersect the edge of the image are not closed *(Figure 6)*. For this reason, polygons that intersect the edge of the image were disregarded in terms of further processing. It could not be assumed these polygons represent the complete extent of the underlying image objects. Therefore further processing and subsequent matches between these polygons and database objects would have been a waste of resource and may have produced inaccurate results. To remove these polygons, an algorithm was implemented that searched all pixels on the edge of the image. When black pixels on the edge of the image were encountered, these were turned white and the value of the pixel at a right angle to this pixel location was also assessed. The discovery of a black pixel at this location indicated the possible existence of a polygon that intersected the edge of the image. This triggered a recursive search of the adjacent pixels. The pixel at the centre of this search was turned white. In the event that more black pixels were discovered among the adjacent pixel they became the centre of new searches and were also turned white. This process resulted in the polygon being eaten away until the search discovered no more black pixels or the edge of the image was encountered. Upon reaching the edge of the image or failing to discover further black pixels this function terminated and the search along the edge of the image continued at the location where it ended. Figure 7 demonstrates the effect of removing all polygons that intersected the edge of the image by this process.

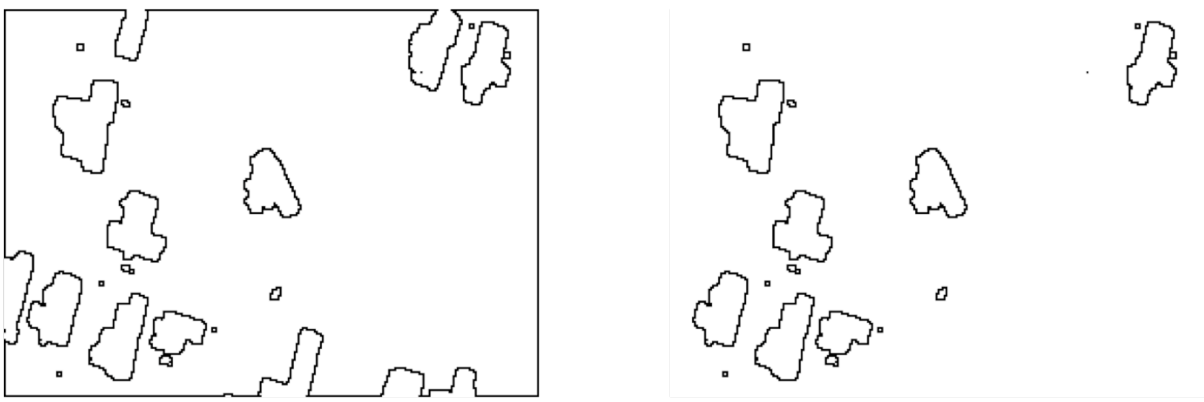


Figure 7 shows the result of the edge detection algorithm used (left) and the effect of removing all polygons that intersect the edge of the image (right).

4.2.4 Polygon Extraction

The extraction of the pixel coordinates for each polygon created during the edge detection process was problematic in that the resulting coordinate sequence needed to be ordered and represent only the outer most points of the polygon. Figure 8 shows a polygon following the edge detection process. Unordered extraction of every pixel location within each polygon as a coordinate pair would result in multiple redundant vertices, as well as multiple self-intersections *(Figure 8).*

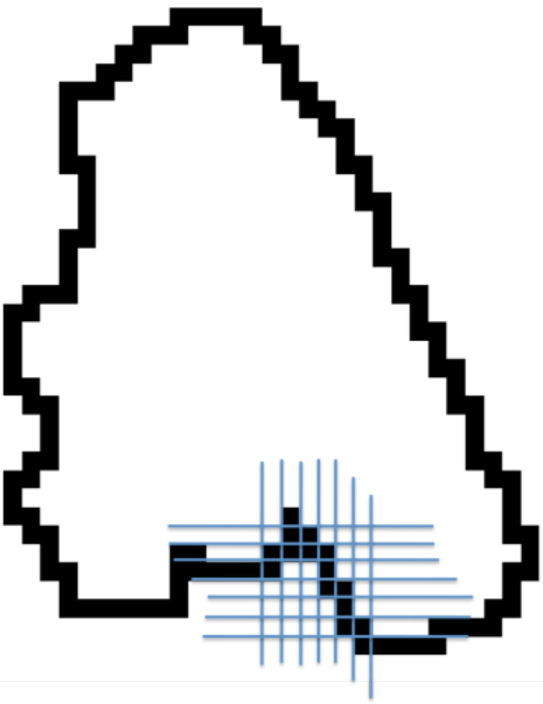


Figure 8 illustrates the relatively complicated form of polygons resulting from the edge detection process. The grid at the bottom of the polygon highlights the individual pixel locations that the polygon consists of. This grid also highlights the fact that the boundary of the polygon is several pixels wide in places. The polygon extraction process focused on extracting the boundary of each polygon as represented by the outer most pixels as an ordered sequence of coordinates.

For this reason boundary coordinates were collected in an ordered fashion, beginning with a pixel by pixel search of the image. During this search if a black pixel was encountered the pixel coordinate was recorded and turned white before the area adjacent to that pixel was searched. Searching in a consistent anticlockwise direction meant that the resulting coordinate sequence was ordered in a consistent direction. Having found the first pixel associated with a polygon, the search area could be greatly reduced by excluding adjacent pixels already considered during the initial pixel by pixel search of the entire image *(Figure 9)*.

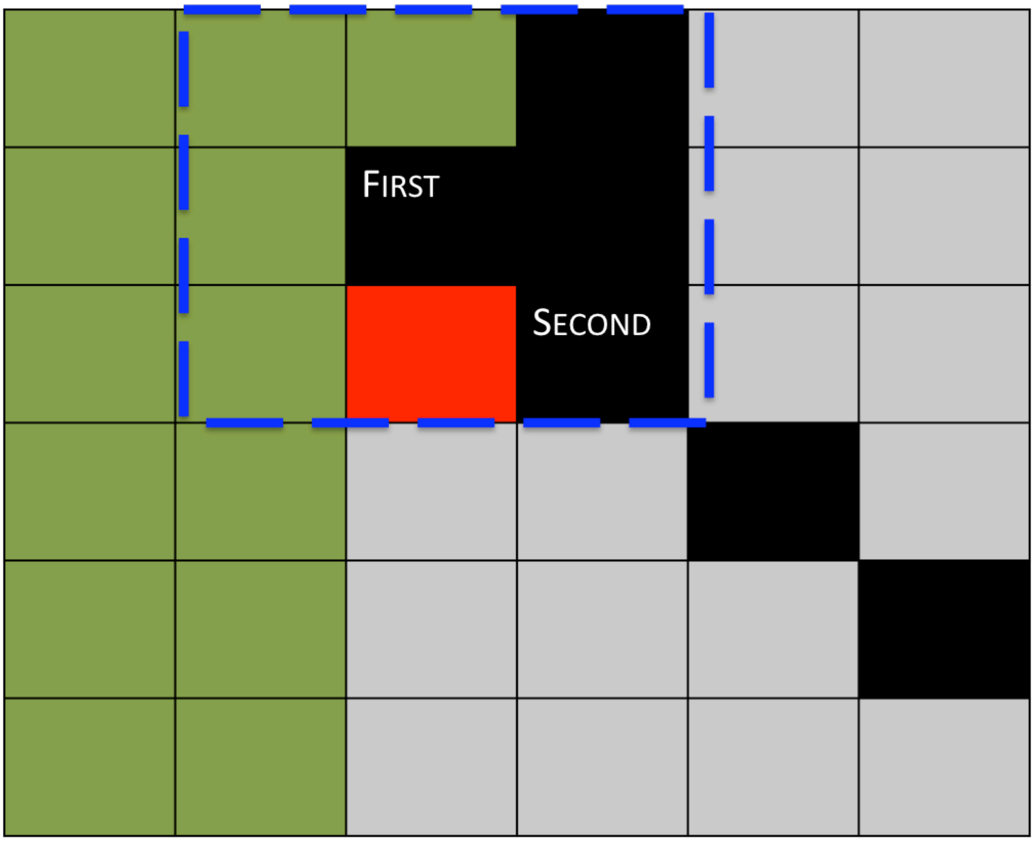


Figure 9 represents a portion of an image during the polygon extraction stage. Green squares represent pixels searched during the initial pixel by pixel search of the complete image (the search direction is from top to bottom and from left to right), black squares represent the pixels that make up the edge of the polygon, the pixel labelled first is the first black pixel to be detected during the pixel by pixel search, the dashed blue box contains pixels adjacent to the first pixel, the red square depicts the start position for the search of the pixels adjacent to the first pixel, the square labelled second shows the position of the next black pixel found during the search of the pixels adjacent to the first pixel and grey squares represent pixels that have not yet been considered.

Excluding pixels already searched during the initial search meant that the green pixels in Figure 9 could be ignored. Therefore the start location of the new search from the first black pixel encountered is the red pixel *(Figure 9)*. The technique not only reduced the number adjacent pixels to be searched but also effectively ensured that only the outer most pixels were collected. Figure 9 demonstrates this by showing that the pixel to the right of the first pixel and vertically above the second pixel is not collected as part of the polygon. The algorithm worked in a recursive manner, meaning that each new black pixel discovered within a search of adjacent pixels triggered a new search centred on the newly discovered pixel. During these searches the same previously described technique was used to limit the search space and ensure that pixel locations were recorded in a sequential manner and in a constant direction. Figure 10 illustrates the method in which further searches were conducted.

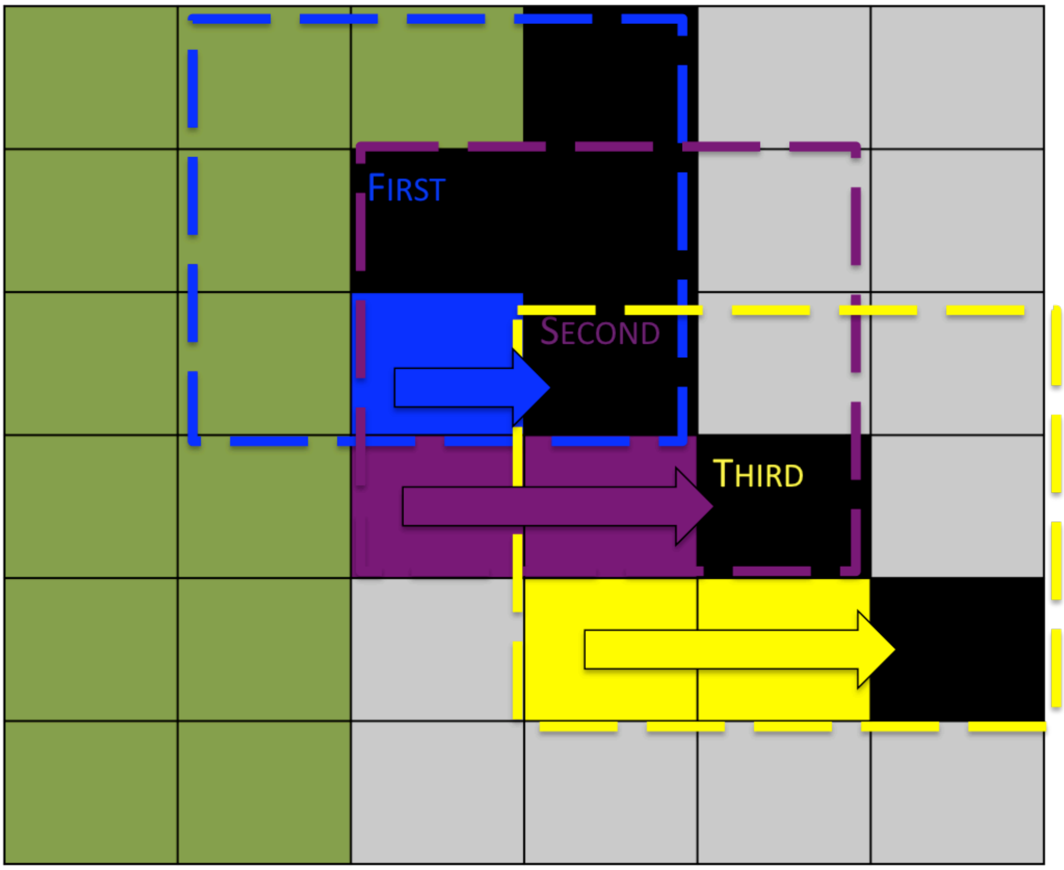


Figure 10 represents a portion of an image during the polygon extraction stage. Green squares represent pixels searched during the initial pixel by pixel search of the complete image (the search direction is from top to bottom and from left to right), black squares represent the pixels that make up the edge of the polygon. The pixels labelled FIRST,SECOND and THIRD have an associated search space symbolised by a dashed line of the same colour, pixels of the same colour within each search space indicate pixels that were searched and arrows of the same colour indicate the direction of the search. Pixels within each search space that are not the same colour as the search space indicate pixels that were either not considered because they were included in pervious searches or a black pixel was found before these locations were reached.

This method meant that searches did not always begin at the same relative pixel location, for example, directly above the centre pixel. Assuming each search had started at the same relative pixel location, each search could be re-searching locations already ruled out during the previous search *(Figure 10)*. For this reason, the start of each search was defined as the first adjacent pixel in an anti-clockwise direction that had not been searched in the previous search. This ensured that the returned coordinates were ordered and represented a polygon that did not self intersect or represent in a multipolygon. Self intersecting or muiltipolygons are invalid inputs into the shape matching algorithm. Polygons were considered valid when the list of coordinates was longer that two coordinate pairs and the search space contained the first coordinate pair, indicating that the current pixel location was adjacent to the first recorded pixel location i.e the polygon was closed. The number of coordinate pairs is important because the search space of the second black pixel found will contain the first discovered black pixel. Without this length requirement, polygons could be considered to be complete prematurely. Upon reaching the end of a polygon this recursive process ended and the pixel by pixel search of the image resumed.

Figure 11 shows the effect of the previously described search algorithm in terms of the resulting shape of the polygon. From Figure 11 it is clear that not all pixel locations have been assigned as polygon vertices. It is important to note that the spatial resolution of the images being processed was 50 cm; therefore the exclusion of these pixels does not represent a significant loss of spatial information or a major change to key geometric attributes. Instead, this process provided the first step in terms of polygon simplification. By excluding these pixels, polygons were simplified and potential processing overhead was avoided. Polygons resulting from this process that contained less than 10 coordinate pairs were not considered for further processing. Objects of this size are unlikely to be adequately represented considering the spatial resolution of the imagery and were therefore not considered for further processing.

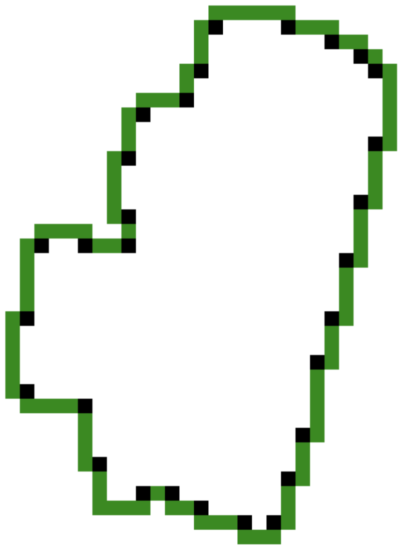


Figure 11 shows the green outline a polygon extracted from an image with the black squares representing pixels that were excluded by the recursive searching process.

4.2.5 Coordinate Conversion

Having extracted potential building polygons from the image, it was necessary to convert polygon coordinates from image space to geographic space. This conversion was achieved using the following equation which describes an affine transformation;

$$X1 = Ax + By + C$$

$$Y1 = Dx + Ey + F$$

Where:

X1 is the calculated Longitude/Easting coordinate

Y1 is the calculated Latitude/Northing coordinate

x is the column number of a pixel in the image counting left to right

y is the row number of a pixel in the image counting from top to bottom

A is the dimension of a pixel in map units in the x direction

B and D are rotational parameters

C and F are translation parameters

E is the negative dimension of a pixel in map units in the y direction

Using the C Sharp GDAL bindings these parameters were calculated from information inherent within each geotiff and each coordinate in a polygon was converted from pixel coordinates to geographic coordinates. Without this conversion, comparisons between image derived polygons and database polygons would have been problematic. Conversion meant that all polygons were comparable in scale and provided a means of assessing distance between polygons and calculating the area of each polygon. Distance between potential matching polygons and area were used as another means of limiting the number of image polygons to be tested against database polygons. If the distance exceeded 30 meters, then the likelihood of a match was deemed improbable and the matching process was terminated. This distance was chosen as it represents the assumed maximum error in terms of the georeferencing accuracy of the images used. The same reasoning was applied to area, if the difference in area between two polygons extended 40 square meters the matching process was considered unlikely to return a positive result and therefore discontinued.

As a result of the thresholding and polygon extraction processes, each image derived polygon consisted of a simplified set of the original pixel coordinates which were converted to geographic coordinates. Thus the geometric centre of each polygon was known in terms of pixel coordinates and geographic coordinates.

4.3 Selecting Polygons For Comparisons

In order to avoid comparisons between every polygon extracted from an image and every polygon within the database, database polygons were selected for comparison based on spatial proximity. Each polygon within the database was spatially indexed using a Quadtree spatial indexing class provided within the NetTopologySuite. The previously described method of converting pixel coordinates to geographic coordinates was used to calculate a polygon representing the spatial extent of each image. The polygon was buffered by 5 meters to allow for the spatial inaccuracies of the direct georeferencing process. Database polygons contained within the spatial extent of an image were able to be efficiently retrieved due to the implementation of the spatial indexing. This provided an efficient method of limiting the number of comparisons between image derived polygons and database polygons to those that were most likely to return spatial accurate results.

4.4 Shape matching

Shape matching consists of the following sub-processes *(Figure 12);*

- Polygon simplification
- Matching within a distance tolerance
- Accounting for spatial offset
- Accounting for rotational differences

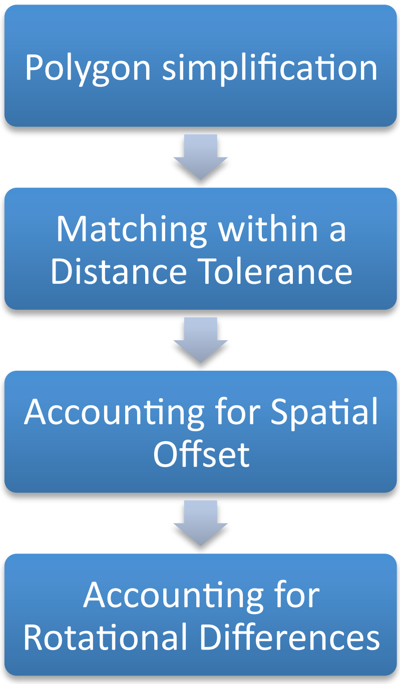


Figure 12 provides an overview of the processes used to match polygons

These sub-processes are designed to firstly simplify the polygon by reducing the number of vertices, and then compare the geometry of the polygon against other polygons. Matches between polygons are successful if geometries can be matched within a distance tolerance. The use of a distance tolerance as the matching criteria requires the spatial offset between polygons to be considered. Assuming that differences in spatial alignment between the data sets may result in rotational differences between individual polygons, the rotation of individual polygons is also accounted for.

4.4.1 Polygon Simplification

Deriving vector representations from raster data as described by the thresholding and polygon extraction process, results in polygons that may contain multiple redundant vertices *(Figure*

11). In an effort to further reduce the amount of processing resource required, image derived polygons were simplified using the Douglas-Plucker algorithm available within the NetTopologySuite. Using simplified polygon representations during all further operations reduced the volume of data be considered during each processing step and provided a more realistic representation of the real world objects.

4.4.2 Matching Within A Distance Tolerance

Converting image features to vector objects results in the vector objects appearing stepped *(Figure 11).* This means that if matches between image derived vector objects and the existing vector building data are to be based on shape, image derive polygons need to be simplified or matched with a distance tolerance. In the absence of either of these two approaches, geometries could be logically equivalent but physically different. This physical difference between image and database polygons could also result from inaccurate digitising of the building outlines. Although polygon simplification algorithms may provide part of the solution, if exact matches were needed to relate one geometry to another, it is unlikely that matches between the two datasets would have been found. Given the unlikelihood of obtaining exact matches between image and database polygons, this research focuses on matching geometries within a distance tolerance. The fundamental idea behind this approach is to match geometries in spite of small differences between them. The Hausdorff Distance provides a formal definition of difference between two geometries. The Hausdorff Distance is the distance between two point sets, which is calculated by finding the maximum distance of a point set to the nearest point in the other point set (Vivek and Sudha, 2007). Therefore matching within a distance tolerance could be formally defined by the rule; geometries match if the Hausdorff Distance between both geometries and their boundaries is less than the distance tolerance (Blasby, et al. 2009). The open source Java library; Java Conflation Suite, is designed to provide various tools to help solve technical difficulties arising during the process of unifying two distinct datasets. This library provides a solution to matching vector objects within a distance tolerance by way of the following algorithm;

Geometry A matches geometry B if;

A is contained in buffer(B, distance tolerance)
AND B is contained in buffer(A, distance tolerance)
AND boundary(A) is contained in buffer(boundary(B), distance tolerance)
AND boundary(B) is contained in buffer(boundary(A), distance tolerance)

It is important to note that A and B are both polygon geometries and the boundaries of these geometries, as returned by the boundary() function, are linestrings. The difference being that polygons are closed geometries whereas linestrings are open. Buffering a polygon by a positive distance results in a similar geometry of larger dimensions with the inverse being true when a negative buffer is calculated *(Figure 13)*. Buffering a linestring also results in a polygon *(Figure 13)*. Figure 14 illustrates the aforementioned shape matching algorithm.

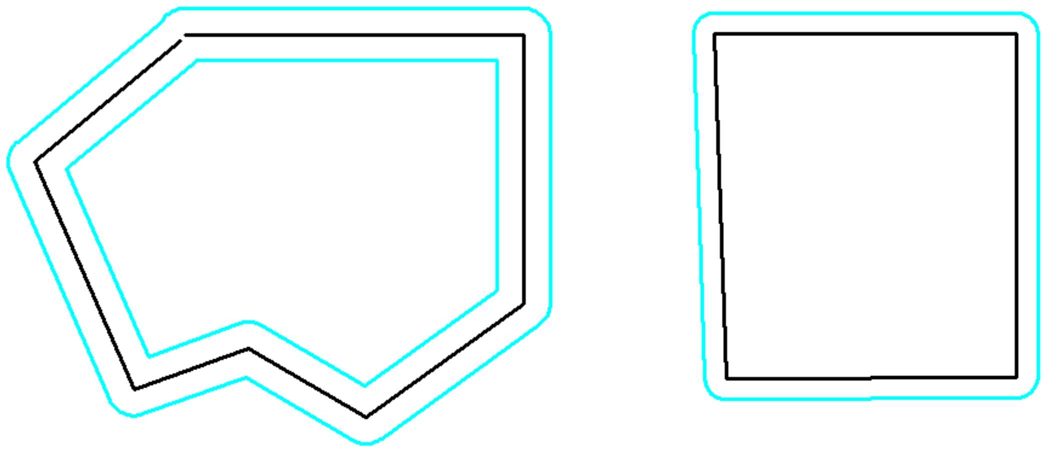


Figure 13 demonstrates the result of buffering a linestring (left) and a polygon (right). The black lines represent the original geometry and the blue lines represent the resulting buffer.

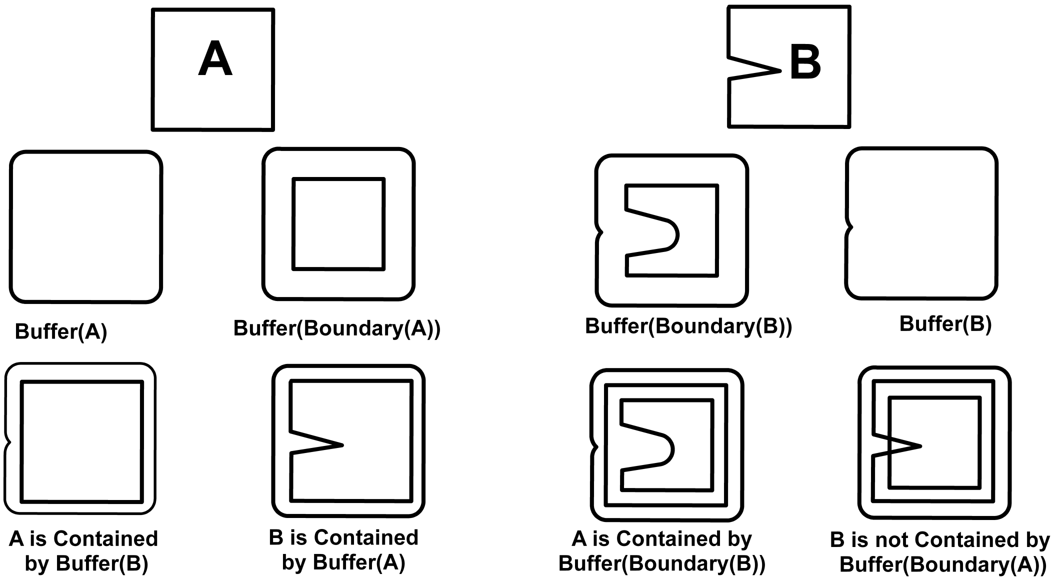


Figure 14 illustrates the way in which buffers are used to check if geometries match within a distance tolerance. In this situation both geometries are contained within the geometry buffers, however, B is not contained within the buffer of boundary A.

The previously described algorithm provided a conceptual basis for the approach used during this research. No equivalent of the Java Conflation Suite exists for C Sharp, therefore, a comparable methodology was constructed with one notable exception; instead of buffering linestrings, positive and negative polygon buffers were used to give the same result. The NetTopologySuite defines a polygon as a linearring, meaning that the start and end coordinate pair are the same. The boundary of a polygon is defined as a linestring, which is not a closed geometry and as such is returned as an ordered list of unique coordinate pairs i.e the last coordinate pair is removed. This meant that if the exact methodology outlined within the Java Conflation Suite was followed, buffering the boundary of a polygon would not accurately represent the geometry of the polygon *(Figure 15).* This would result in the previously described matching algorithm returning a negative result for all matching attempts.

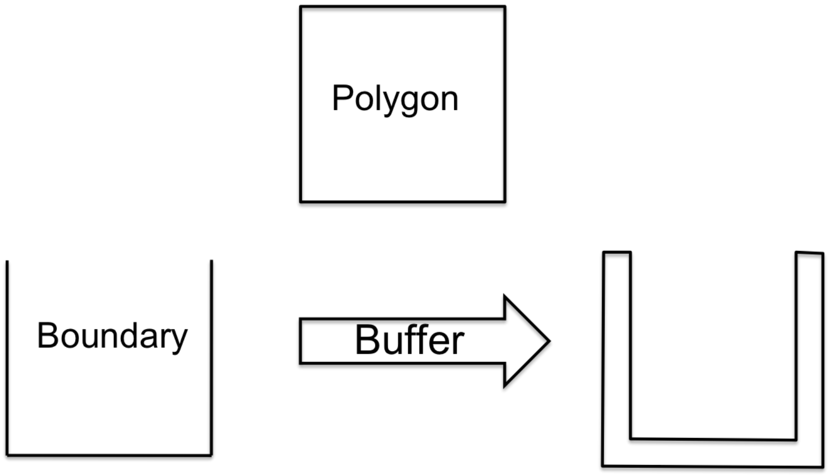


Figure 15 illustrates the effect of buffering the boundary of a polygon as defined by the NetTopologySuite. Buffering the boundary of a polygon does not result in the accurate representation of the original geometry.

In order to account for this, two options were considered. The first option consisted of defining a new coordinate close enough to the first coordinate of the polygon so that the buffering operation would enclose this new end coordinate and the first coordinate, thus accurately representing the original geometry. The second option was to use positive and negative buffing operations to create two polygons, external and internal to the original geometry *(Figure 16)*. For the purposes of this research the second option was preferred given its conceptual simplicity and ease of implementation.

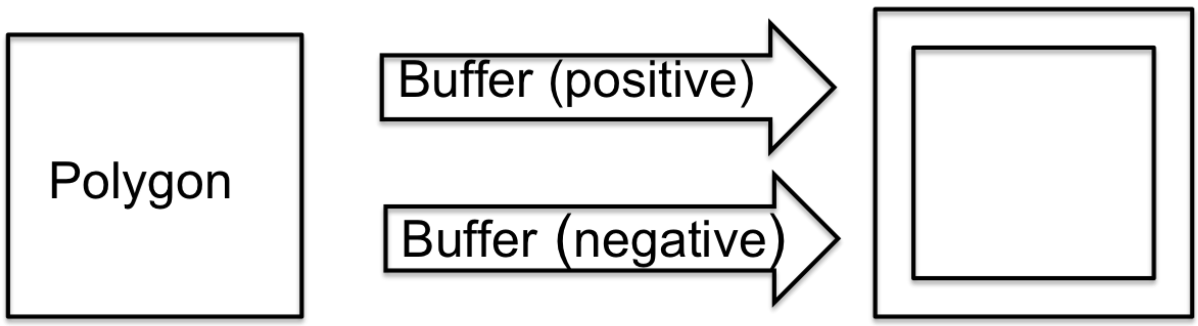


Figure 16 demonstrates the effect of positive and negative buffering operations carried out on a simple polygon. It is important to note the output of both positive and negative buffering is a separate polygon.

By creating both positive and negative buffers, the two resulting polygons can be used to mimic the containment test using a buffered linestring as suggested by the Java Conflation Suite implementation. The complete shape matching algorithm implemented during this research is as follows;

Geometry A matches geometry B if;

A is contained in buffer(B, distance tolerance)
AND B is contained in buffer(A, distance tolerance)
AND A is contained in buffer(B, positive distance tolerance)
AND A contains buffer(B, negative distance tolerance)
AND B is contained in buffer(A, positive distance tolerance)
AND B contains buffer(A, negative distance tolerance)

The distance tolerance used during this research was two meters. Initial trails suggested that two meters provided an effective trade off between accuracy of matches and the number of matches. Shorter distances resulted in less mismatches between polygons, however the number of matches was greatly reduced, and thus far fewer images were successfully georeferenced. The inverse was also true and resulted in a greater numbers of images being successfully georeferenced, but these were often severely distorted due to the number of mismatches between polygons. Later trials involving larger data sets confirmed the initial results. The optimal distance tolerance is likely to vary with the quality and spatial resolution of the data in combination with the ease with which image features can be accurately segmented.

This algorithm allows geometries to be compared and matched within a given distance tolerance, however, it assumes no rotational differences between geometries and that geometries exist in the same spatial location. In terms of matching image derive polygons with database polygons; it could not be assumed that no spatial offset existed or that no rotational differences were present.

4.4.3 Accounting For Spatial Offset

For the buffering method to work it is necessary for both input geometries to share a common centroid. Without a common centroid, geometries with a spatial offset greater than that of the given distance tolerance would not have been matched, even if they were identical. Spatial offset could occur for a variety of difference reasons. With regard to the images processed during this research, the major contributing factor was inaccuracies in the initial capture of data resulting from the process of direct georeferencing. However, inherent distortion within aerial imagery such as radial displacement could also cause this. Differences in spatial location may also be a legacy of the methods used to capture the database polygons. Before each image derived polygon was tested against selected database polygons, the spatial offset between both centroid locations was calculated and used to offset the image derived polygon. This meant that both polygons now shared the same centroid. Thus any failure to match the two would not be a result of a difference in spatial location.

4.4.4 Accounting For Rotational Differences

As with spatial offset, it could not be assumed that rotational differences between image derived polygons and database polygons were non-existent. To ensure that differences in rotation did not cause logically equivalent polygons to return a negative result when compared, an algorithm was designed to rotate image polygons by one degree increments 25^o anti-clockwise and 25^o clockwise *(Figure 17)*. As both polygons share a common centroid (see Section 4.4.3), the image polygon was rotated around this point. The following equation describes this process;

$$X1 = xc + (Cos (D) * (xp - xc) Sin (D) * (yp - yc))$$
$$Y1 = yc + (Sin (D) * (xp - xc) + Cos (D) * (yp - yc))$$

Where;

X1 is the rotated Longitude/Easting value

Y1 is the rotated Latitude/Northing value

xc is the Longitude/Easting value of the polygon centroid

yc is the Latitude/Northing values of the polygon centroid

xp is the Longitude/Easting value of the polygon vertex

yp is the Latitude/Northing value of the polygon vertex

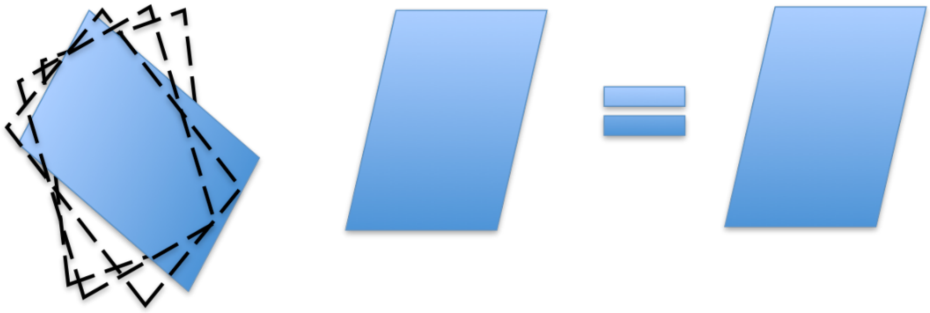


Figure 17 illustrates the rotation process. The dashed black lines represent progressive rotations of the underlying blue rectangle to the right. This figure demonstrates the utility of the rotating the polygon, it is clear that in the absence of the rotation process a negative result would have been returned by the matching process.

The decision to rotate each polygon no more than 25^{o} in both clockwise and anti-clockwise directions follows the assumption that although the georeferencing of each image is not completely accurate, the performance overhead far exceeds the potential benefits of any further rotations. Also, buildings may have a similar shape but different orientation, by rotating each polygon a limited number of degrees the chances of matching an image derived polygon to a database polygon of similar shape but significantly different orientation are reduced. After each incremental rotation, both polygons are compared. This process was repeated until the comparison between the two polygons returned a positive result or the polygon had be rotated 25^{o} in both a clockwise and anti-clockwise directions.

4.5 Filtering Multiple Matches Between Polygons

The database of building outlines used during this research was created by manually digitizing building outlines from georeferenced aerial imagery with a spatial resolution of approximately 50 cm *(Figure 1)*. Although 50 cm resolution aerial imagery is sufficient to allow a skilled operator to collect an accurate polygon representation of a building outline, it also means that the collected polygon may not uniquely identify the building it represents. This assumption is further justified by the fact that building shapes commonly consists of rectangular features that, when simplified, appear similar. Given the occurrence of multiple similar shapes within the database and that the matching process used a distance tolerance of two meters, matches between individual image derived polygons and multiple database polygons could occur. Matches between polygons were only attempted if the distance between the two centroids was less the 30m, in an effort to restrict matches between polygons to those within a similar spatial context. However, it is still possible for the image thresholding and contouring process to generate polygons that either do not represent a building, or represent only part of a building. Polygons resulting from this may be comparable to those in the database. This situation has the potential to result in matches between non-building objects or partial representations of buildings and building outlines within the database. To combat this, two filtering algorithms were designed.

4.5.1 Filter One

Filter one considered the offset of all matches between image derived polygons and database polygons. Offset in this sense was defined as the difference between the pixel coordinates of image derived polygons and the geographic coordinates of database polygons. This approach follows the assumption that the spatial offset between all correctly matched image derived polygons and data base polygons is consistent and that the number of correct matches is greater than the number of incorrect matches *(Figure 18)*.

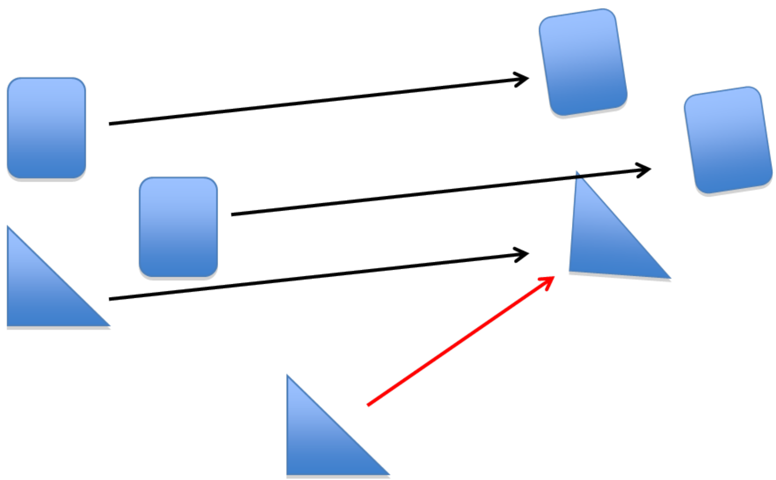


Figure 18 depicts a number of shapes with arrows indicting geometric matches between shapes. Black arrows indicate matches between shapes that have a similar spatial offset, the red arrow indicates a spatial offset that is inconsistent with those represented by the black arrows. Matched pairs of image derived polygons and data base polygons with similar spatial offsets were considered more likely to represent reality than matched pairs with inconsistent spatial offsets.

This was done by calculating the spatial offset between every matching pair of polygons. Firstly, matching pairs of polygons were filtered to ensure that multiple copies of the same pair did not exist. Given the potential variability between multiple correctly matched polygon pairs, the sorting process needed to allow for slight differences in spatial offsets caused by variations in centroid locations. Offsets were considered the same if the variation did not exceed 2 meters in either the x or the y component. The offset of each pair was then used to select other pairs with the same offset. This process creates multiple copies of the same matching pair of polygons, these copies were removed. The end product was a list of unique matching pairs of image derived polygons and data base polygons with the same spatial offset between pixel coordinates and geographic coordinates.

4.5.2 Filter Two

Filter two calculated the error for each GCP and the mean error of all GCPs. As C Sharp provides limited support in terms of mathematical functionality, filter two consisted of custom software written in the Python programming language. This software received a list of GCPs from C Sharp and calculated an affine transformation based on this list. The transformation meant that for any given pixel location a corresponding geographic coordinate could be calculated. Consequently, for each pixel location in a GCP two geographic coordinates existed, one defined by the transformation one within the GCP. The distance between these two coordinates is the error associated with the given GCP. If the mean error was greater than three, the Python software removed the GCP with the highest error and recalculated the transformation. This was continued until the mean error was less than three or only four GCPs remained. This process provided an effective means of filtering outliers from potential GCPs. The remaining GCPs were written to a text file which was then read by C Sharp.

4.6 Correcting The Original Image

The previously described filtering process produced objects consisting of the following information;

The centroid of the image derived polygon (expressed as a pixel location)
The centroid of the database polygon(expressed in geographic coordinates)

The GDAL provides a number of utility programmes designed to carry out fundamental operations on geospatial data. The geotifcp utility programme produces a georeferenced image when supplied with a header file containing specific information relating to the desired geographic coordinate system of the output image and pixel locations with their corresponding geographic coordinates. The result is a geotif which has been rectified using an affine transformation calculated from the given pixel locations and geographic coordinates. Matches between image polygons and database polygons meant that the centroid of the image polygon (expressed as pixel values) could be related to the centroid of the corresponding database polygon (expressed as geographic coordinates). Pixel values and their corresponding geographic coordinates were written into a header file and, along with a copy of the original image, input into geotifcp. Geotifcp then produced a geotiff version of the original image, georeferenced according to the transformation derived from input pixel locations and corresponding geographic coordinates.

5. Results and Discussion

The following presents the results of applying the previously described methodology to a sample of 394 directly georefereced aerial thermal images. Of these images 141 were corrected during the shape matching procedure. The methodology implemented during this research resulted in two data sets, one containing the original directly georeferenced images and the other containing the images that had been successfully corrected via the shape matching process. In order to make comparisons between the two data sets, a third set of manually georeferenced images was created. This manually created set consisted of copies of those images re-georeferenced as a result of matches between image features and database objects. It is important to note that this research assumes manually georeferenced imagery to produce the best possible result in terms of spatial accuracy, and effectively provides the "truth" against which images resulting from this methodology are tested.

The first step during the process of manually georeferencing these images involved a conversion from geotiff to .bmp format. The choice of .bmp format was arbitrary, the conversion provided a simple means of ensuring that the spatial information stored in the header of each geotiff was removed and therefore did not influence the process of manual georeferencing. The use of different data sources during this process would have rendered any comparison between the manual and automated methods of georeferencing meaningless. For this reason, only the database of building outlines was used as a reference for the manual georeferencing process. Manual georeferencing was done in ArcGIS Desktop 9.3 and a world file was created for each image. World files are simple txt files that contain the transformation parameters calculated for an individual image. These parameters define a first order polynomial transformation based on the given GCPs. The transformation parameters contained in each world file define a model, which can be used to calculate the corresponding geographic coordinate for any given pixel location. This enabled direct comparisons between image data sets to be made.

For the sake of simplicity, direct, corrected and manual image data sets, will refer to directly georeferenced images, images corrected during the implementation of the methodology described in section 4 and images created via manual georeferencing, respectively. The following attributes were calculated for each within the direct and corrected data sets, as will be discussed in detail below:

- Difference in rotation compared to the same manually georeferenced image
- Accuracy of GCPs
- Accuracy of randomly selected pixel locations
- The Standard Distance Deviation of GCPs associated with each image
- The Mean Error associated with each image.

These attributes resulted from the application of custom software written in C Sharp and Python programming languages using the C Sharp GDAL bindings and the NetTopologySuite, and provide a quantitative measure of the accuracy for comparing the direct, corrected and manual image data sets.

5.1 Calculating The Rotational Difference Between Images

Each georeferenced image has the same resolution, 263 * 193 pixels, and a unique set of transformation parameters contained within a corresponding world file. By transforming pixel locations 0,0 and 263,0, into geographic coordinates, it was possible to represent the long axis's of each image as a line between two real world locations. In this way lines could be defined using the same pixel coordinates but the differing transformation parameters associated with each image. Comparisons between images were made by defining a line for each image, calculating the slope of each line and using the resulting slope values as input into an equation to determine the acute angle between the two lines *(Figure 19)*.

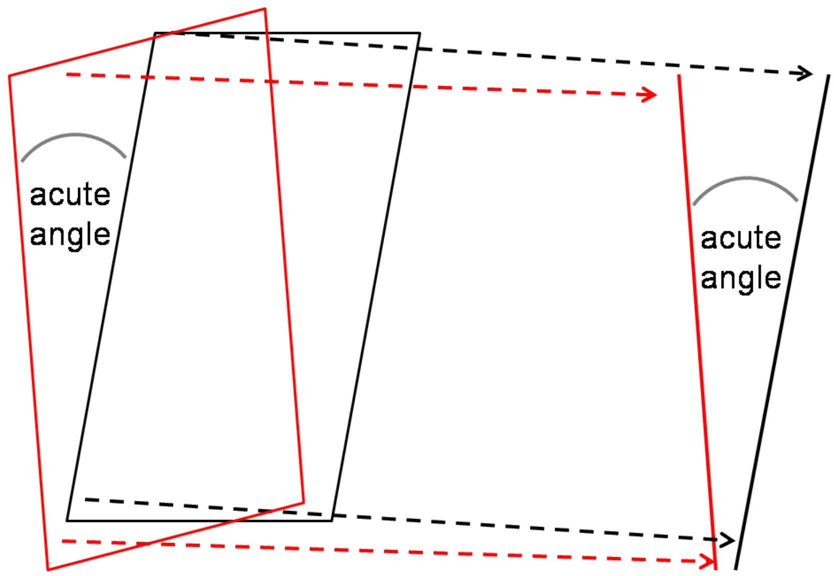


Figure 19: The red and black rectangles represent the polygon outlines of two images. Although the polygons are exact replicas of each other they have been rotated differently. By isolating the same longitudinal axis of each polygon, it is possible to first calculate the slope of each line and then the acute angle between the two lines. The acute angel is the difference in rotation between the two lines and therefore the two polygons.

The slope of a line is defined as;

$$M = (y1 - y2) / (x1 - x2)$$

Where m is the slope of the line
Y1 is the first y value
Y2 is the second y values
X1 is the first x value
X2 is the second x value

The acute angle between two lines is defined as;

$$A = \arctan(M2 - M1/ 1 + M1 * M2)$$

Where

M1 is the slope of a line

M2 is the slope of a line

M2 > M1

By calculating the acute angle between the two lines representing the longitudinal axes of two images, it is possible to gain a measure of the relative difference in rotation between the two images *(Figure 20).*

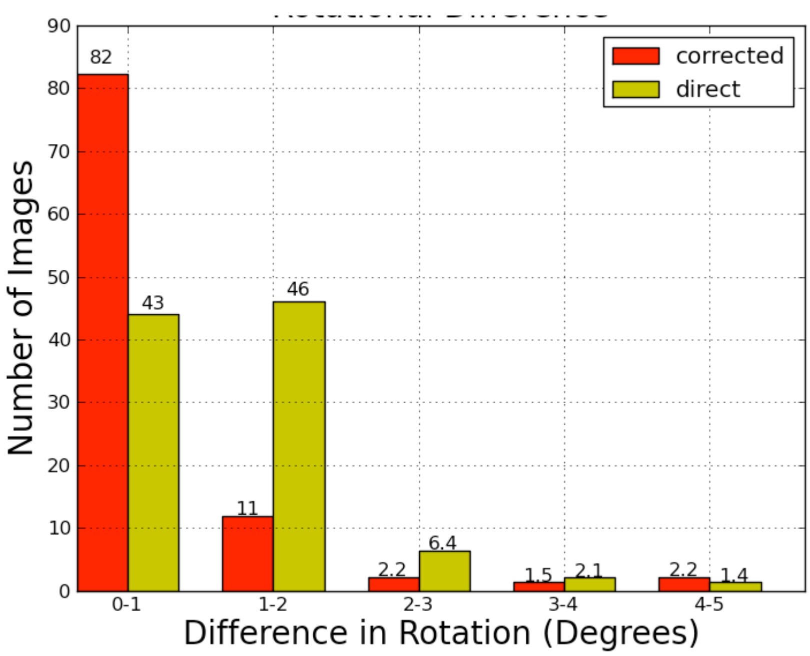


Figure 20 displays the calculated rotational difference between the corrected and the manual data sets and between the direct and manual data sets.

The assessment of rotational difference between both the corrected and direct data sets is not conclusive in terms of which methodology ultimately produced the most accurate results *(Figure 20)*. 83 percent of the corrected images have a rotational difference of less than 1 degree compared to 43 percent of the direct images *(Figure 20)*. This suggests that the shape matching process produces images with an orientation that is more likely to reflect reality, however, this in itself does not automatically mean that these images have a greater spatial accuracy. For example the long axis's of two images may run parallel to each other, however, the existence of a spatial offset between the two images would not be detected by an examination of the rotational differenced. Thus, the orientation may be accurate but not the spatial location. Figure 20 also suggests that the 25^{o} of rotation in both clockwise and anticlockwise direction performed during the shape matching procedure may have been significantly more than was required (see Section 4.4.4). Assuming that features within images are rotated proportionally to the image, Figure 20 suggests the image derived polygons that are matched after being rotated by more than 5^{o} are likely to produce inaccurate results.

5.2 Calculating The Accuracy Of GCPs

Direct images and corrected images were in geotiff format, meaning that their associated GCPs are stored as header information within the image. Using the C Sharp GDAL bindings, header information contained within each geotiff could be accessed. This meant that a complete list of the pixel locations and their corresponding geographic coordinates were available. For each GCP within a geotiff, the pixel location was used as input into the transformation of the corresponding manually georeferenced image. This produced two geographic coordinates for each pixel location stored in the GCPs. The distance between these two points provides a measure of spatial accuracy for an individual GCP. Individual distances were recorded for both the direct and corrected data sets. Comparisons between the original images and those successfully georeferenced by the shape matching process suggests a considerable improvement in the spatial accuracy of individual GCPs within images produced during the course of this research *(Figure 21)*. 93 percent of GCPs produced by the shape matching process are within 5 meters of the same GCP in the manually georeferenced data set. By comparison, only 14 percent of GCPs produced by direct georeferencing are within 5 meters of the same GCP in the manually georeferenced data set *(Figure 21)*.

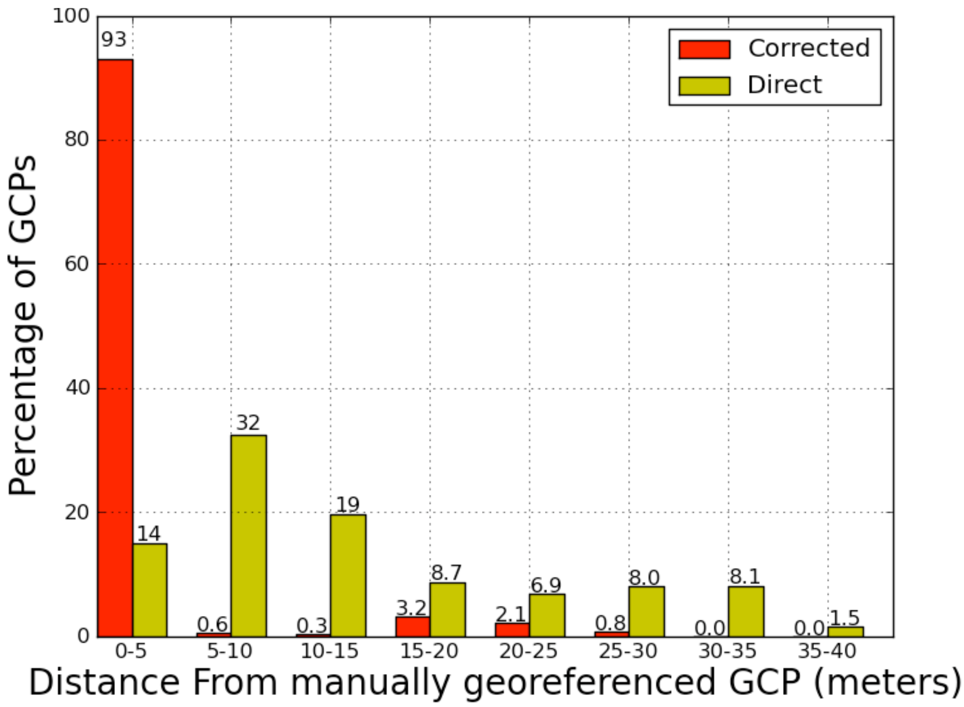


Figure 21 depicts the percentages of GCPs from both corrected and direct data sets that fall within the given distances of their actual locations as defined by the manual data set.

Figure 22 focuses exclusively on GCPs that have an accuracy of 5 meters or less and demonstrates that the 79 percent of the corrected GCPs are within 2 meters of their actual location, whereas this is only true for 3.4 percent of direct GCPs. The matching process is based on matching polygons within a distance tolerance of two meters, it is likely that the distance tolerance used will affect the accuracy of the GCPs. It is important to note that GCPs were not explicitly recorded for the manually georeferenced images. Comparison between manually georeferenced images and corrected images are based on the use of transformation parameters stored in the associated world files to calculate the geographic coordinates of a given pixel location. The transformation defined by these parameters is essentially a model of best fit, it may not strictly represent the GCPs used to define it. Therefore, a portion of the error seen in the corrected and direct GCPs can be attributed to the transformation used to define the geographic location of pixels within the manually georeferenced data set. As for the 7 percent of corrected GCPs with spatial inaccuracies of greater than 5 meters, it is likely that they result from positive matches between logically different geometries. The amount of these mismatches could conceivably be reduced by reducing the distance tolerance, this is also likely to reduce the over amount of accurately matched points. However, misrepresentation of a building outline due to over-hanging vegetation, adjoining buildings or poor contrast between buildings and the surrounding environment for a given threshold level may still result in matches between different objects. While corrected GCPs tended to be more accurate than direct GCPs they were also less numerous *(Figure 23).*

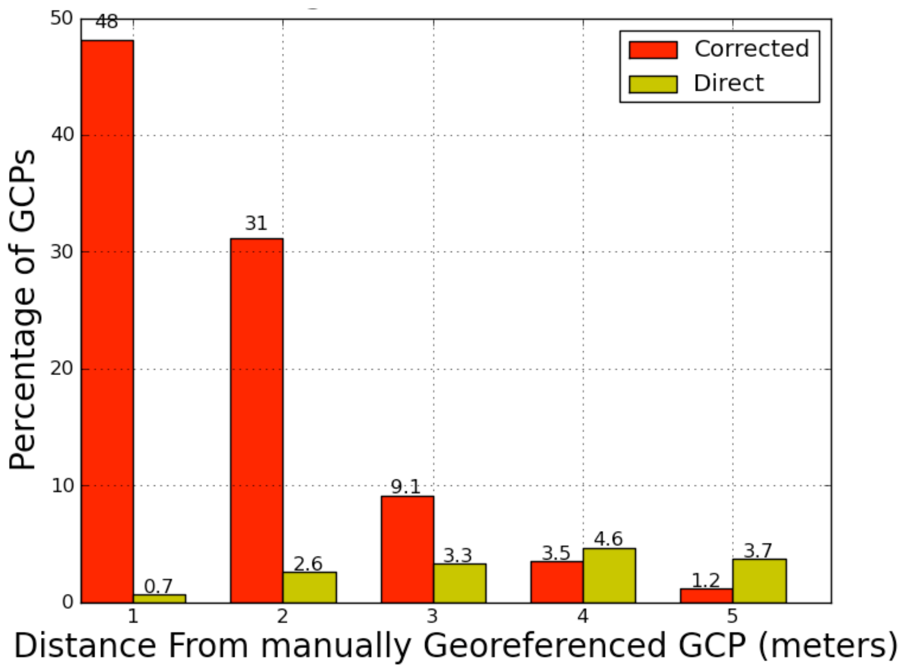


Figure 22 depicts the percentage of GCPs from both Corrected and Direct data sets that are within 5 meters of their actual locations as defined by the manual data set.

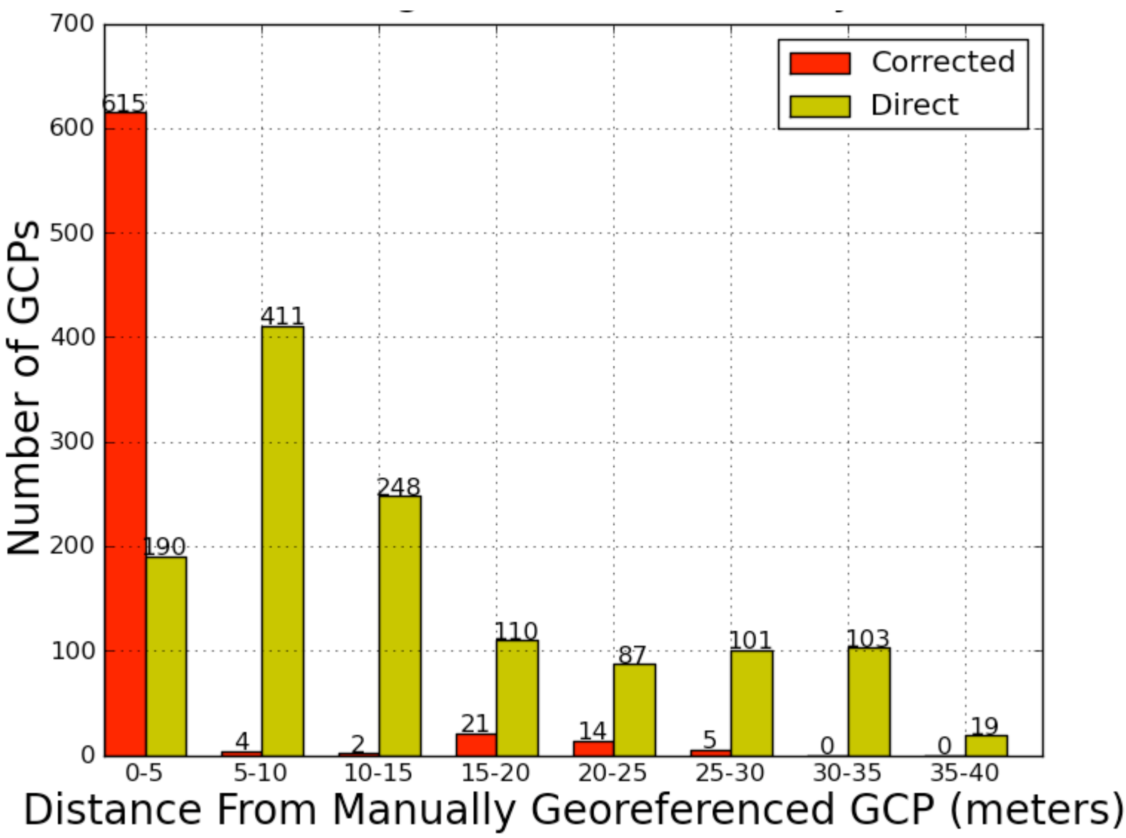


Figure 23 depicts the histogram of GCP accuracy for both the corrected and direct data sets

5.3 Calculating The Accuracy Randomly Selected Locations

Increased spatial accuracy of individual GCPs, in theory increases the overall accuracy of the associated transformation and therefore the accuracy of the geographic coordinates calculated by the transformation for a given pixel location. Figures 21, 22 and 23 suggest that the shape matching process is capable of significantly increasing the spatial accuracy of individual GCPs. In order to gauge the overall accuracy of the calculated transformations spatial accuracy was measured at 100 locations within each image in both corrected and direct data sets. This was achieved by producing 2 groups of 100 randomly varying values between 0-263 and 0-193 for every image in both the corrected and direct data sets. As previously stated all images used and produced during this research had a resolution of 263 * 193 pixels and an associated world file. By sequentially selecting one value from each group of random values, random pixel locations were constructed. Geographic coordinates were calculated for these pixel location using the transformation parameters contained within the world file of a given image. The same pixel coordinates were used to calculate geographic coordinates for the corresponding image from the manual data set. The distance between the two geographic coordinates represents the spatial error. This process was carried out for each image in the corrected and direct data sets and produced 14100 data entries for each data set. Data entries for the corrected data set were then divided into groups depending on the number of GCPs in their associated image. This created 6 distinct groups of spatial accuracies given that corrected images had 3,4,5,6,7 or 9 GCPs *(Figure 24)*.

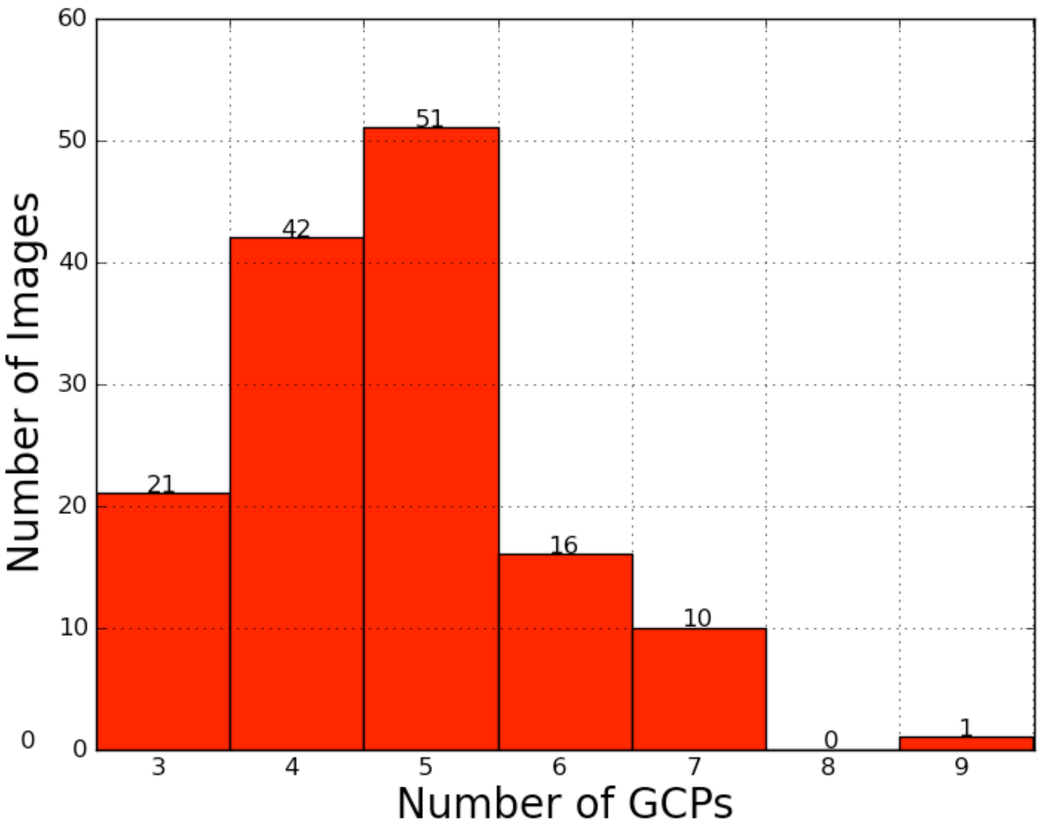


Figure 24 depicts the number of corrected images with 3,4,5,6,7 and 9 GCPs.

All direct images consisted of 9 GCPs, however, given the variation in spatial accuracy of individual GCPs within the direct data set, it is highly likely that individual images also vary in terms of spatial accuracy. For this reason, comparisons are made between corrected images with a given number of GCPs and the corresponding images from the direct data set. Comparisons are visualised by way of Box plots, where the red line represents the median spatial error, the blue rectangle represents the distribution of 25% to 75% of the data and the extents of the dashed lines extending from the blue rectangle represent the minimum and maximum errors.

Figures 25 - 30 depict box plots of the spatial errors associated with corrected images with 3,4,5,6,7 and 9 GCPs and the spatial errors of the same images from the direct data set.

Figure 25 suggests that corrected images with 3 GCPs, as a group, have a considerably greater variation in spatial accuracy compared to the same images in the direct data set. The variation in spatial accuracy within the direct images appears relativity consistent between Figure 25 and 26, whereas, variation in the corrected images decreases dramatically. Figure 26 demonstrates that the maximum error in the corrected data set is greater than that of the direct data set, however, it also shows that 75% of corrected values have an accuracy of approximately 0 – 12 meters compared to 75% of direct values which have an accuracy of approximately 0 - 17. The methodology described in section 4 requires a minimum of 3 GCPs to georeference an image. It is possible to accurately georeference an image with only three accurate, well distributed GCPs, however, it would seem that in the context of this research, 3 GCPs is insufficient. The significant increase in accuracy seen between corrected images containing 3 GCPs and corrected images containing 4 GCPs suggests that 4 may be a more appropriate minimum number *(Figure 25, 26)*.

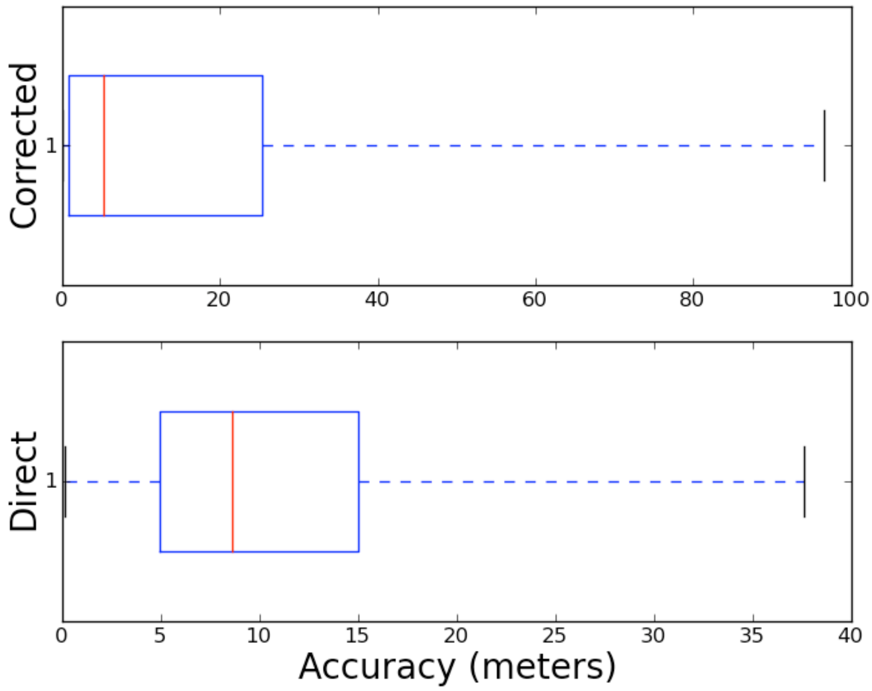


Figure 25 depicts the spatial error of the 21 corrected images with 3 GCPs and spatial accuracy of the same images from the direct data set.

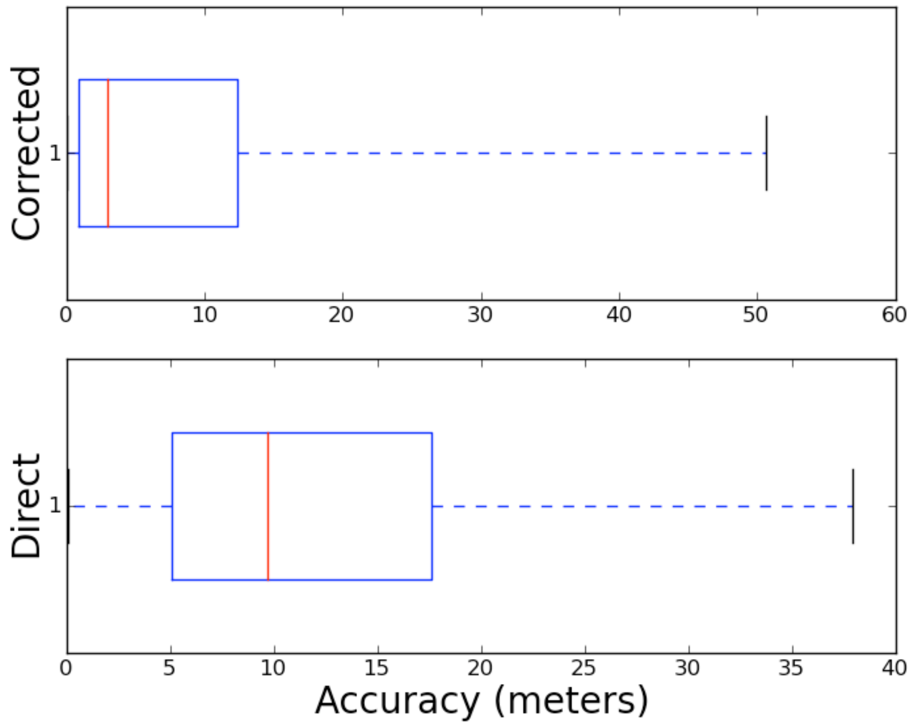


Figure 26 depicts the spatial error of the 42 corrected images with 4 GCPs and spatial accuracy of the same images from the direct data set.

Corrected images containing 5 and 6 GCPs show maximum errors that are comparable to the maximum error in the direct data set *(Figure 27, 28)*. Given that 75% of error in these corrected data sets is between 0 and 2.5 meters, it seems likely that significantly larger errors may be associated with individual images and not evenly distributed throughout all images *(Figure 27, 28)*. It seems improbable that an individual transformation model that locates 75% of randomly selected pixel locations within approximately 2.5 meters of their real world locations would produces spatial errors of up to 29 meters in the remaining 25%. Assuming individual images are responsible for the range of errors, examination of the polygon outline of each image may provide a means of filtering these images during the processing stage. If a polygon outline differs significantly in size or dimension from the majority of other polygon outlines it could be assumed that the associated image contains significant spatial error, and thus identified for removal.

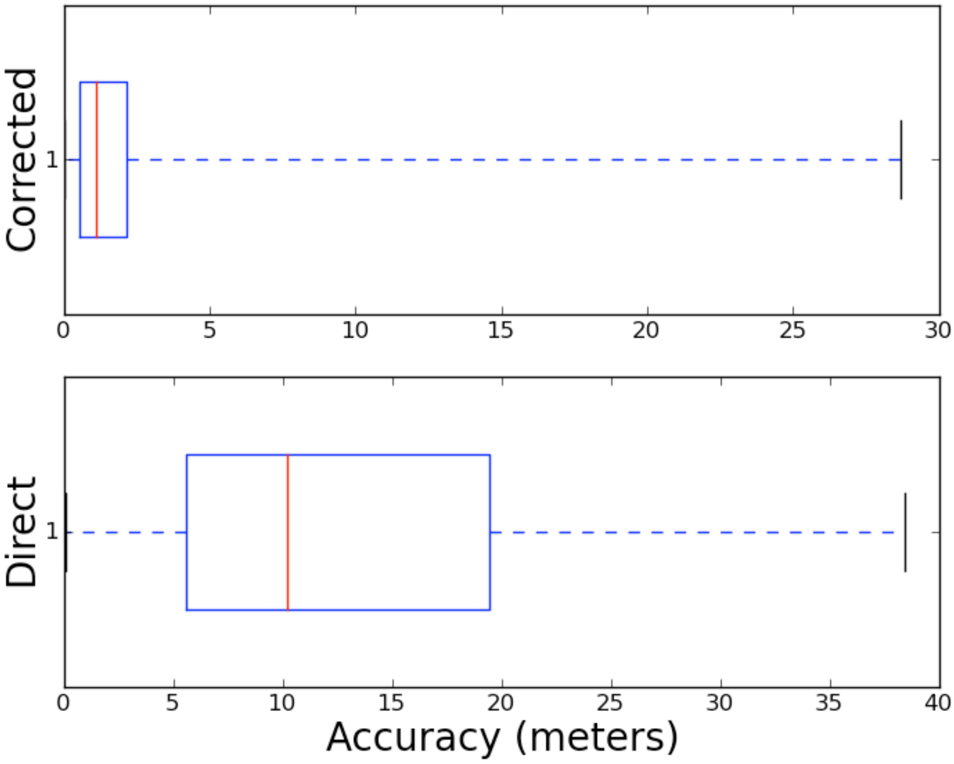


Figure 27 depicts the spatial error of the 51 corrected images with 5 GCPs and spatial accuracy of the same images from the direct data set.

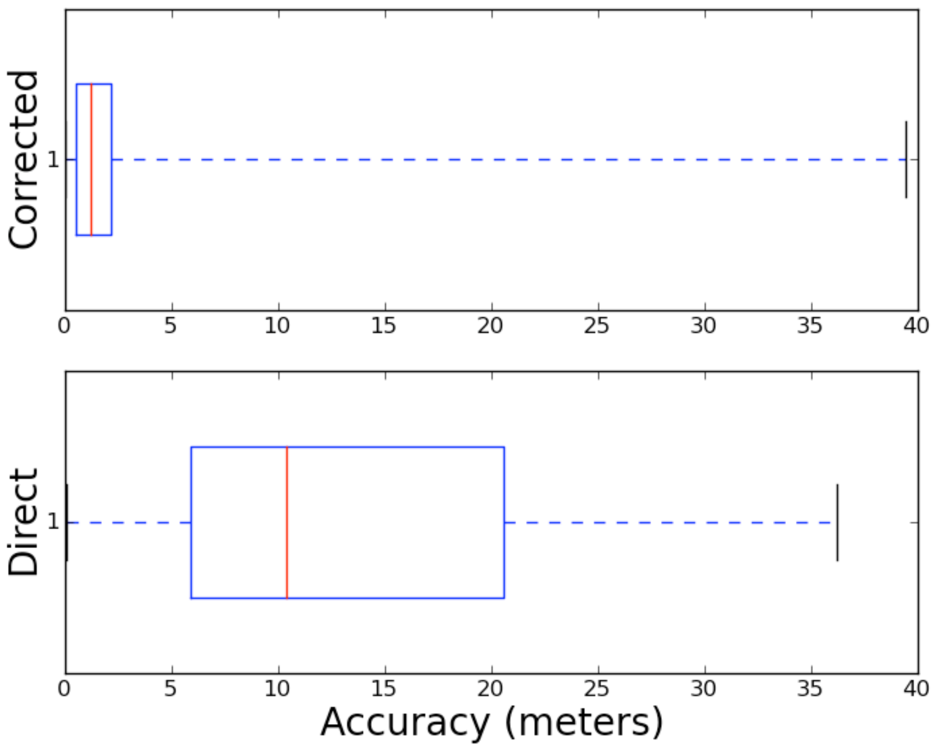


Figure 28 depicts the spatial error of the 16 corrected images with 6 GCPs and spatial accuracy of the same images from the direct data set.

Figures 29 and 30 demonstrate that images with 7 and 9 GCPs offer a significant improvement in spatial accuracy compared to the same images in the direct data set. With the maximum error not exceeding 3 meters and 75% of all data having a spatial error of less than 1.25 meters, the utility of these images in terms of analysis in combination with existing spatial data is greatly increased.

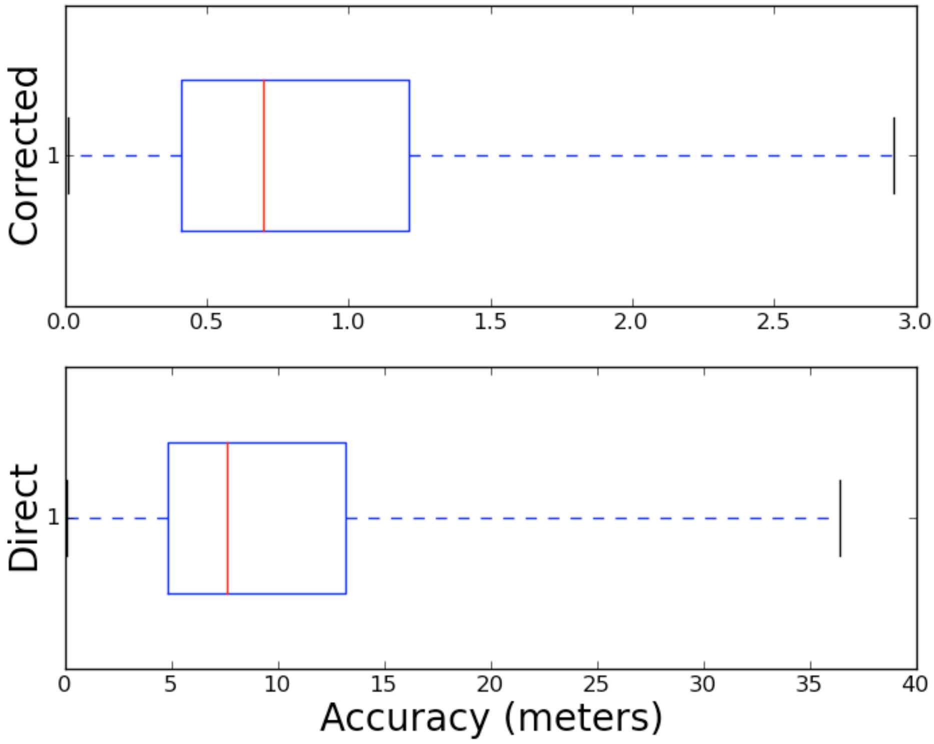


Figure 29 depicts the spatial error of the 10 corrected images with 7 GCPs and spatial error of the same images from the direct data set.

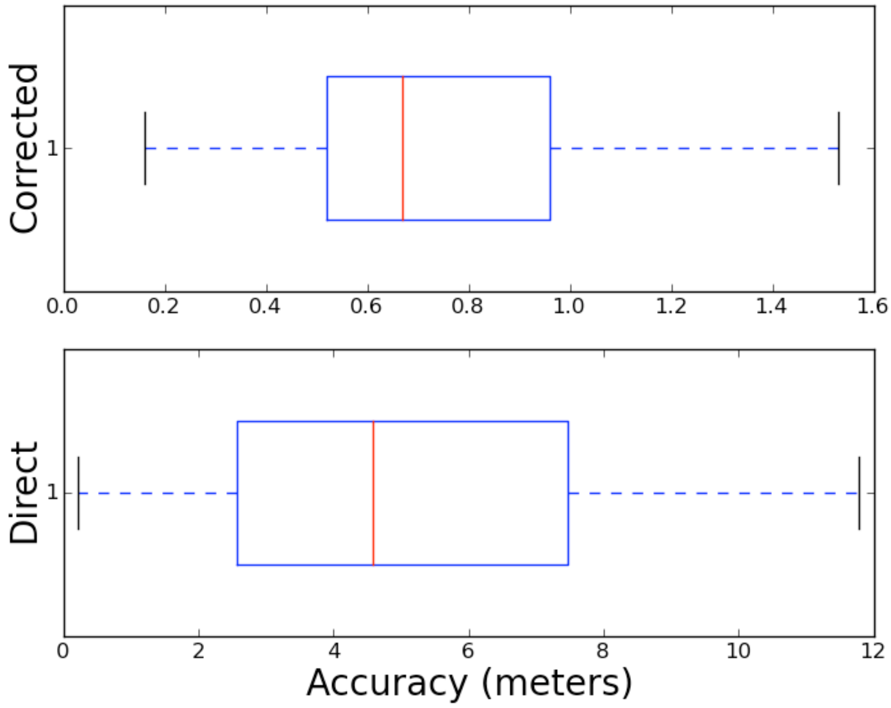
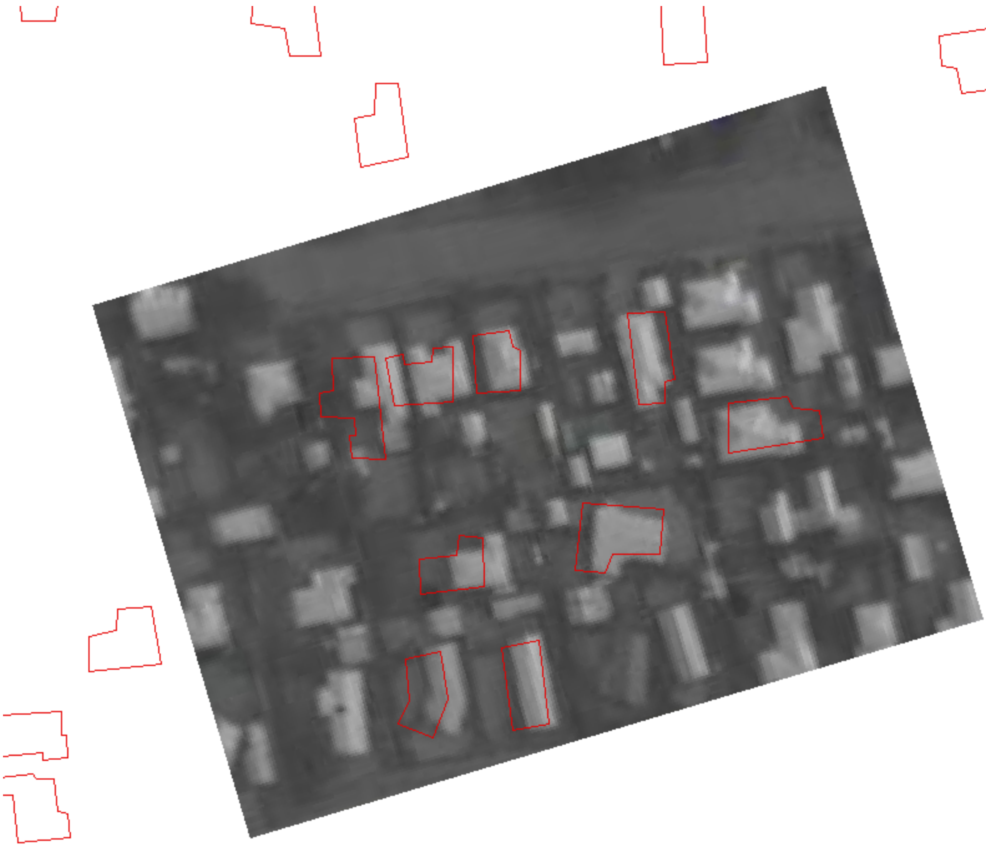


Figure 30 depict the spatial error of the single corrected image with 9 GCPs and spatial error of the same image from the direct data set.

Figure 30 offers a direct comparison between the single corrected image consisting of 9 GCPs and the corresponding direct image. This is a clear illustration of the potential gains in spatial accuracy as a result of the shape matching process. The spatial accuracy of the corrected image varies from approximately 0.18 – 1.58 meters whereas the spatial accuracy of the direct image varies from approximately 0 – 12 meters. It is also important to note that 75% of the one hundred random pixel locations from the corrected image have sub meter accuracy, whereas, 75% of the same pixel locations are accurate to within approximately 7.5 meters in the direct image. Figure 31 offers a visual comparison between both images.



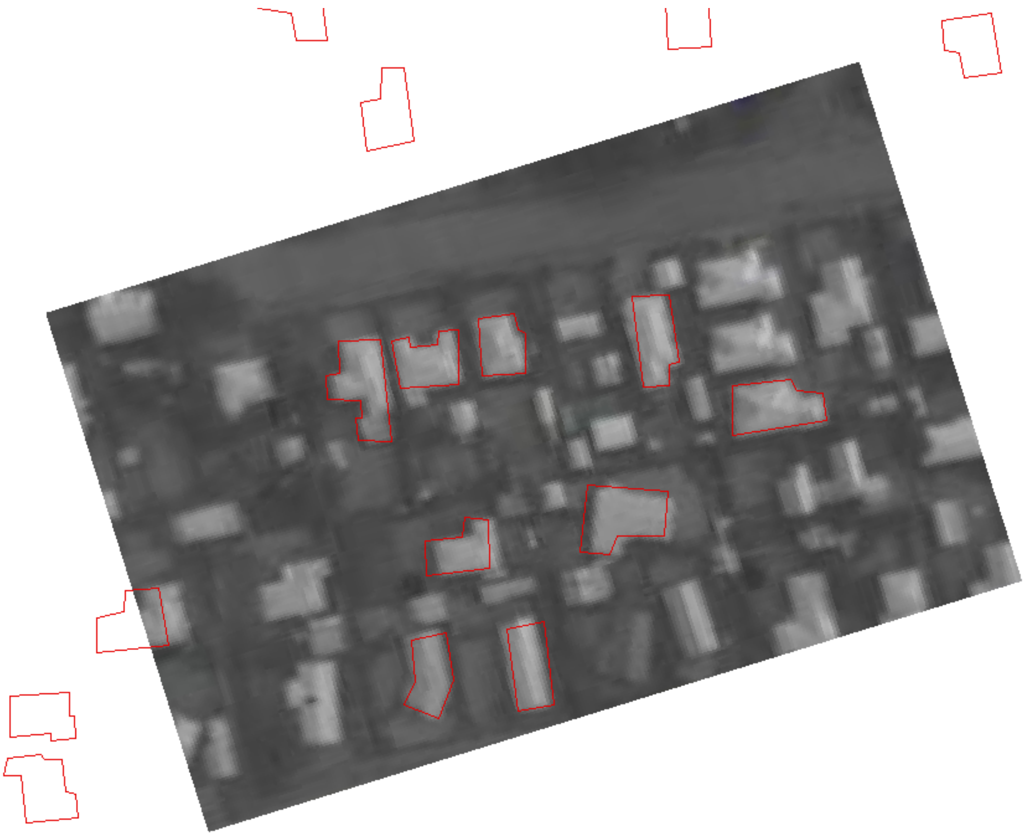


Figure 31 offers a visual comparison between the direct image (above) and the corrected image (below). The spatial accuracies of these images are depicted by box plots in Figure 30.

The previous comparisons between corrected images and the same direct images demonstrate that a significant amount of variation exists in the spatial accuracy of corrected images. While variation is also present in the direct data set, it is of a much smaller scale. Figures 21, 22 and 23 suggest that the shape matching process produces GCPs that are considerably more accurate but also less numerous than those existing within the direct images. The number of GCPs is important in that a higher number of GCPs increases the likelihood of GCPs being widely distributed throughout an image. Widely distributed GCPs result in transformations that more accurately reflect the real world locations of pixels throughout the entire image as opposed to those in close proximity to individual GCPs. This may explain the greater variation in spatial accuracy in the corrected data set, despite the fact that individual GCPs were more accurate. This assumption is supported by Figures 28, 29 and 30, where the decrease in spatial variation with increasing numbers of GCPs is apparent.

5.4 Calculating The Standard Distance Deviation Of GCPs

The Spatial distribution of GCPs throughout an image influences the overall accuracy of the transformation calculated for a given image. It is usually considered preferable to have highly distributed GCPs, with corner locations seen as optimal positions. Figure 32 demonstrates the result of poorly distributed GCPs and their affect on the resulting georeferenced image. Despite the fact that the GCPs in Figure 32 are accurately located, poor distribution results in inaccurate image rotation.

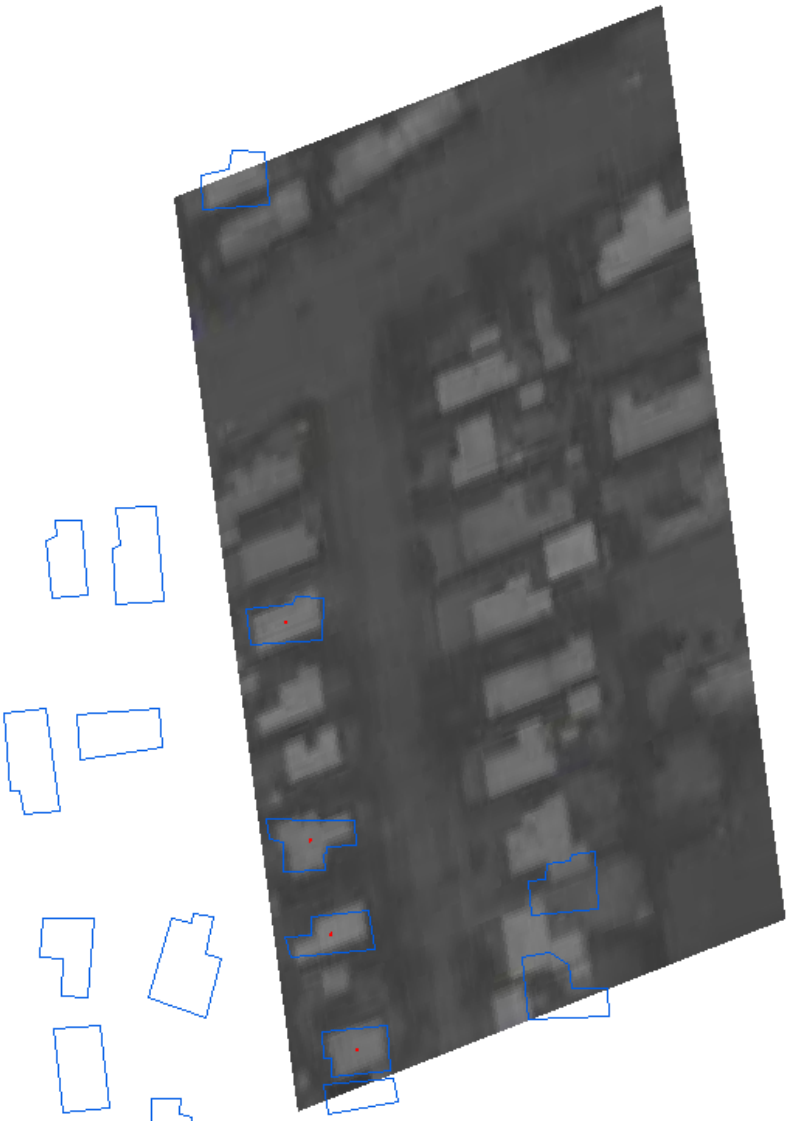


Figure 32 displays the effect of poorly distributed GCPs. Blue polygons represent building outlines and the four red dots within four of the building outlines represent GCPs used to georeference the image. The image is noticeably distorted; this is a direct result of the GCPs distribution.

In an effort to obtain a measure of the distribution of GCPs throughout each geotiff in both the corrected and direct data sets, the Standard Distance Deviation (SDD) was calculated and recorded for each image. The SDD is defined as the standard deviation of the distances from each GCP pixel location to the centroid of all the GCP pixel locations. In practical terms this was achieved by creating a multipoint geometry from all GCP pixel locations within the regular raster of a given image. The centroid of this multipoint defined the centre of all GCP pixel locations. The distances between each GCP pixel location and the centroid pixel location were calculated and recorded. The SDD is the standard deviation of these recorded distances. The SDD is therefore a measure of GCP distribution within the image raster. The larger the SDD, the greater the pixel distance between GCPs and the better the distribution. Although perhaps not the last word in spatial distribution, the SDD does provide a useful, single figure measurement for spatial distribution of points. While this measure does not provide a definitive answer in terms of the spatial accuracy of the image, it is another means of assessing the difference between directly georeferenced images and those produced the implementation of this methodology. Figure 33 demonstrates variation from 10 – 40 pixels in the SDD of the corrected data set while the direct data set varies from 35-50 pixels. This suggests that the procedures used to calculate GCPs for the direct data set produce relatively evenly spaced GCPs. This is supported by the fact that every image in the direct data set contains 9 GCPs. In contrast, the shape matching process is reliant on the number of matches between image features and data base objects to create GCPs, therefore the distribution of GCPs is less likely to be inconsistent. This assumption of inconsistency is supported by Figure 33.

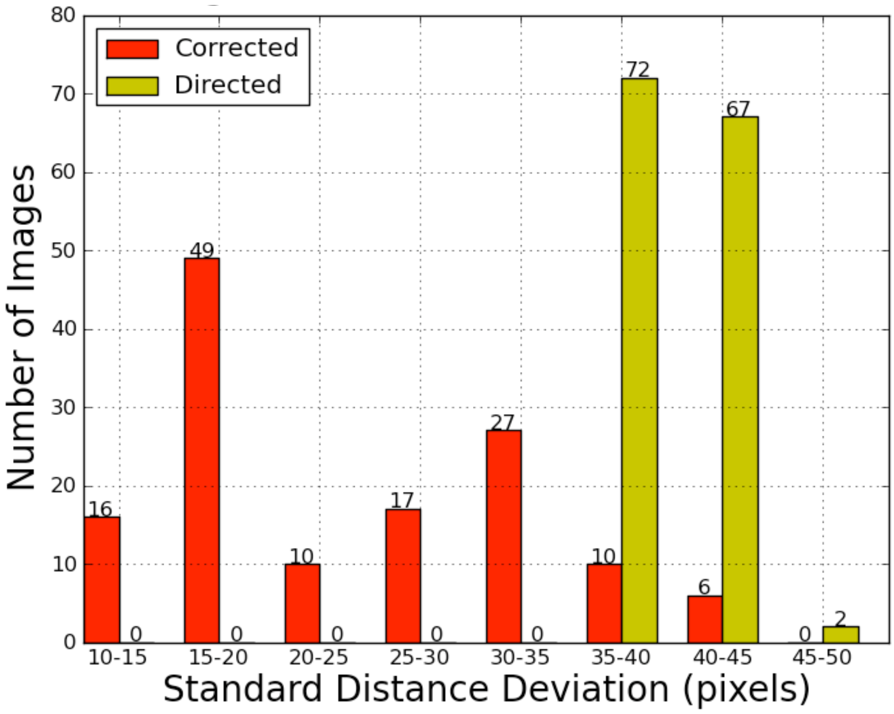


Figure 33 depicts the variation in distribution of GCPs within corrected and direct data sets

It is interesting to note that despite high variability in the distribution of GCPs and the distributions of less than 25 pixels in the majority of images, based on the previous results the correct data set appears to be more spatially accurate than the direct data set. This suggests that the influence of other factors such as the number and accuracy of GCPs in the determination of overall spatial accuracy. It would also seem that relatively accurate results can be obtained with poorly distributed GCPs. This being the case, it is highly likely that the GCPs in the corrected data set are well distributed in terms of their x and y values. Distribution with regard to differences in the x and y axis' are not expressed using the SDD.

5.6 Calculating The Mean Error

The mean error for each image was established by calculating the distance between the geographic location of each GCP and the geographic location as defined by the transformation parameters for the image using the GCP pixel location as input. This distance was calculated for each GCP in an image and represents the error associated a given GCP. The Mean Error for an image is thus the mean of these distances. The Mean Error is often used as a means of providing a measure of precision during the process of georeferencing. However, this does not present an intuitive measure of spatial accuracy, but rather a measure of how well a GCP fits in the current transformation as defined by existing GCPs. Spatially accurate GCPs may easily be considered inaccurate if the model already consists of uniformly inaccurate GCPs. The corrected data set appears to be more accurate in all other aspects, however the direct data set has a higher number of images with a lower Mean Error *(Figure 34)*.

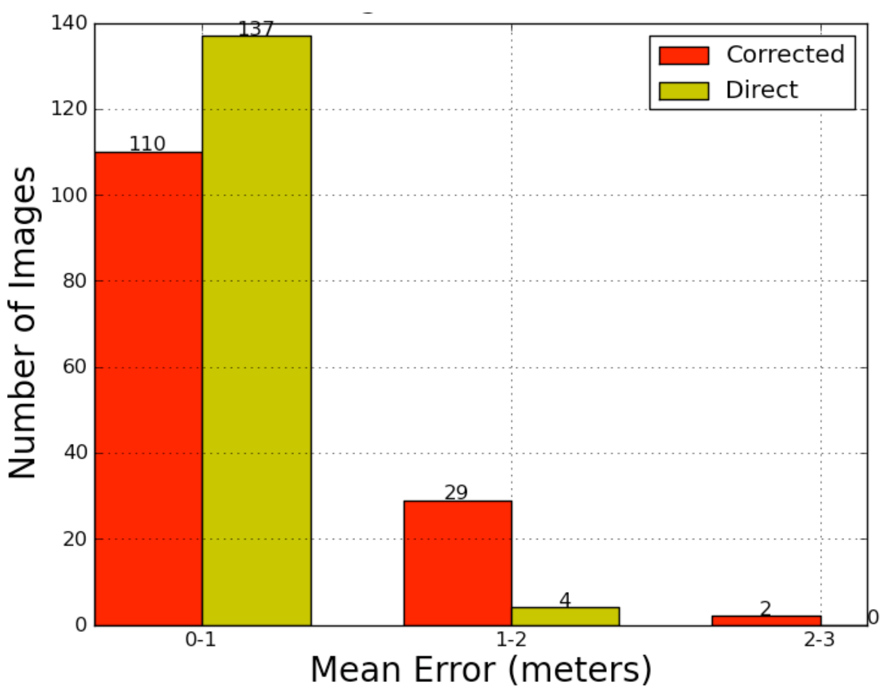


Figure 34 depicts a histogram of image and their mean error

With regard to the corrected data sets it is important to remember that the methodology greatly increased the likelihood of a low mean error. This is probably also the case with the direct data set, as the mean error and error associated with individual GCPs provides an excepted method of filtering outliers from the transformation model (Rousseeuw and Leroy, 1987).

6. Conclusion

The following offers general conclusions that can be drawn from the implementation of shape matching to correct the georeferencing of directly georeferenced images, highlights the limitations of this research and suggests areas for future research.

6.1 Conclusion

The results of this research suggest that image features can be accurately related to database objects based on matching geometries within a distance tolerance. Matches between image features and data base objects for a given image can be exploited to define GCPs for a given image, thus georeferencing the image. This process is capable of significantly improving the spatial accuracy of directly georeferenced thermal aerial imagery. The process of direct georeferencing calculates GCPs for images based on sensor location and physical properties at the time the image was taken, combined with topographic knowledge of the scene being imaged. Applying the direct georeferencing process to the capture of aerial imagery greatly increases the volume of data able to be captured and processed within a given time frame. However, the direct georeferencing process is error prone and the spatial accuracy of the results may limit the utility of the images produced. While, options such as bundle adjustment and refinement through trial and error are available, they both rely heavily on the availability of precise position and attitude data, repeated field campaigns and overlapping imagery. In the absence of these resources, manual effort is required. The methodology described in Section 4 offers the ability to increase the spatial accuracy of directly georeferenced aerial imagery without the need for significant manual intervention, precise position and attitude data, repeating flights or overlapping imagery. This methodology is conceptually simple, considers scale and rotation as well as accounting for spatial offset and makes use of fundamental GIS concepts that exist as pre built tools in modern GIS thus aiding implementation for users without specialist programming skills.

6.2 Limitations

It is important to note that, while the results of this research suggest that the methodology is a valid means of automatically correcting directly georeferenced imagery, the implementation was greatly aided by use of thermal imagery. As previously stated, the thermal imagery used was captured during a warm summer's day and thus the thermal reflectance of buildings was relatively strong. This meant that buildings could be adequately identified as features distinct from their surrounds. It is unlikely that image segmentation will be as easily achieved in the majority of image data sets. Effective image segmentation and the extraction of the segmented image objects as a vector representation is key to this methodology.

Images used during this study were captured over level terrain with no significant distortion caused by high rise buildings. Complex terrain or high rise buildings may produce distortions within imagery, meaning that the extracted image features may not be matched to the database equivalent due to geometric distortion.

This research also assumes the availably of a data base containing spatially accurate vector representations of buildings for the area of interest. The resolution of the available image data is also an important factor when considering the viability of this approach. In order for this methodology to be considered, the resolution of the image data set used must be such that buildings are able to be extracted to vector representations that are comparable to their logical equivalent within the data base.

Given that this research used matches between the geometry image features and data base objects, this methodology may not be appropriate for urban areas that contain limited variation in terms of building shape, size and orientation.

6.3 Future Research

The successful implementation of an adaptive thersholding process would avoid the need to apply multiple threshold values to a single image, thus reducing the amount of duplication in terms of extracted shapes and the subsequent matching attempts. However, it is recognised that the use of thermal imagery in terms of the application of this methodology, represents a very specific case and future applications are likely to require individually tailored algorithms to identify image features.

This research focuses on matching individual geometries; however, successful matches of these geometries result in multiple geometries that share topological relationships with each other. These relationships describe the pattern of the complete set of geometries. While, spatial offset between matched pixel polygons and data base polygons was considered, other topological relationships could also be considered. Pattern matching is already a well established field of research (Schuurman, et al. 2006). Matching objects within a distance tolerance may provide a means of validating the inclusion of individual objects that construct a given pattern. These patterns of objects originating from images could then be matched against patterns of objects found within a data base. In this way, the selection of GCPs would not just rely on individual matches, but also on matches between multiple geometries and their topology.

For the methodology applied during this research to be successful given a dense data base containing surveyed building footprints as opposed roof outlines, it may be necessary to consider aggregating building vector representations. It is likely that building footprints will represent buildings as individual shapes, however, aerial imagery may display buildings with close or touching roofs to be a single object. By aggregating such buildings within a data base into a single polygon, the matching process may be able to match a group of buildings in an aerial image to the corresponding group of buildings in a data base.

This research considers each image in isolation, no effort is made to utilise GCPs found in an image with regard to the previous image or the following image, despite the significant overlap that exists between successive images. The extent to which images may overlap will vary between data sets, However, if overlap between images does exist, it seems likely that this could be exploited to achieve greater spatial accuracy than possible by considering each image as an individual. One approach might involve creating a mosaic of several overlapping images based on matches between image features and then searching the mosaic for image features to match against data base objects. This would mean that individual images within the mosaic could benefit from the GCPs unique to the surrounding images also within the mosaic.

This research used a first order polynomial transformation to define relationships between pixel locations and the corresponding geographic locations. Future implementations of this methodology might consider using higher order polynomial transformations as a means of improving the spatial accuracy of GCPs.

This research provides a custom automated method of matching image features to vector objects for the purpose of correcting the georeferencing of directly georeferenced aerial imagery. This method was applied specifically to a data set of thermal images and a vector data set consisting of manually digitised building outlines. Analysis of the results of this application suggest that matching image features and vector objects within a distance tolerance is an appropriate means of relating pixel locations to geographic coordinates for the purpose of automatically georeferencing images. The implemented method has demonstrated the ability to significantly improve the spatial accuracy of directly georeferenced aerial thermal images.

7. References

ENVI 4.5 documention 2009.

Baltsavias E and Hahn M (2000). "Intergrating Spatial Information and Image Analysis: One Plus One Makes Ten." IAPRS, ISPRS **33**: 63-74.

Baltsavias E P (2004). "Object extraction and revision by imagery analysis using existing geodata and knowledge: current status and steps towards operational systems." ISPRS Journal of Photogrammetry & Remote Sensing. **58**: 129-151.

Benz U C, Hofmann P, et al. (2004). "Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information." ISPRS Journal of Photogrammetry & Remote Sensing **58**: 239-258.

Bernier T and Landry J (2003). "A new method for representing and matching shapes of natural objects." Pattern Recognition(36): 1711-1723.

Binaghi E, Brivio A P, et al. (1999). "A Fuzzy set-based accuracy assessment of sort classifcation." Pattern Recognition Letters(20): 935-948.

Blasby D, Davis M, et al. (2009). GIS Conflation using Open Source Tools.

Bonnefon R, Dherete P, et al. (2002). "Gegraphic information system updating using remote sensing images." Pattern Recognition Letters **23**: 1073-1083.

Cheung C K and Shi W (2006). "Positional error modelling for line simplification based on automatic shape similarity analysis in GIS." Computers & Geosciences(32): 462-475.

Cleve C, Kelly M, et al. (2008). "Classification of the wildland-urban interface: A comparison of pixel- and object-based classification using high-resolution aerial photography." Computers, Environment and Urban Systems **32**: 317-326.

Congalton R G (1991). "A Review of Assessing the Accuracy of Classifications of Remotely Sensed Data." Remote Sensing Of The Environment(37): 35-46.

Costa L and Cesar R M (2001). Shape Analysis and Classification. London, CRC Press.

Durieux L, Lagabrielle E, et al. (2008). "A method for monitoring building construction in urban sprawl areas using object-based analysis of Spot 5 images and existing GIS data." ISPRS Journal of Photogrammetry & Remote Sensing **63**: 399-408.

Ehlers M, Gahler G, et al. (2003). "Automated analysis of ultra high resolution remote sensing data for biotope type mapping: new possibilities and challenges." ISPRS Journal of Photogrammetry & Remote Sensing **57**: 315-326.

Erbisch K (2002). "A correction to the Douglas-Peucker line generalization algorithm." Computer & Geosciences(28): 995-997.

Gamba P, Dell'Acua F, et al. (2005). "Urban remote sensing using multiple data sets: past, present and future." Information Fusion **6**: 319-326.

Goncalves N and Aruajo H (2009). "Estimating parameters of noncentral catadioptric systems using bundle adjustment." Computer Vision and Image Understanding(113): 11-28.

Hahn M and Baltsavias E P (1998). Cooperative Algorithms and Techniques of Image Analysis and GIS. ISPRS Symposium on "GIS - Between Visions and Applications", Sturrgart, Germany.

Jacobsen K (2002). Calibration Aspects In Direct Georeferencing Of Frame Imagery. Pecora 15/Land Satellite Information IV/ISPRS Commission I/FIEOS.

Kato Y, Hirano T, et al. (2007). "Fast template matching algorithm for contour images based on its chain coded decription applied for himan face identification." Pattern Recognition(40): 1646-1659.

King V J and Davis C (2007). "A Case study of urban haet islands in the Carolinas." Environmental Hazards(7): 353-359.

Liu Y, Zhang D, et al. (2007). "A survey of content-based image retrieval with high-level semantics." **40**(Pattern Recognition): 262-282.

Loncaric S (1998). "A Survey of shape analysis techniques." Pattern Recognition(31): 983-1001.

Longley P A, Goodchild M. F, et al. (2005). Geographic Information Systems and Science. Chichester, John Wiley and Sons, Ltd.

Mena J.B and Malpica J. A (2005). "An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery." Pattern Recognition Letters **26**: 1201-1220.

Morgado A and Dowman I (1997). "A procdedure for automatic absolute orientation using aerial photographs and a map." ISPRS Journal of Photogrammetry & Remote Sensing(52): 169-182.

Mouragnon E, Lhuillier M, et al. (2009). "Generic and real-time structure from motion using local bundle adjustment." Image and Vision Computing(27): 1178-1193.

Muller M, Kruger W, et al. (2007). "Robust image registration for fusion." Information Fusion(8): 347-353.

Mustafa A (2002). "Fuzzy shape matching with boundary signatures." Pattern Recognition Letters(23): 1473-1482.

Oke T, Ed. (1993). Boundary Layer Climates. New York, Routledge.

Richards J A (1993). Remote Sensing Digital Image Analysis An Introduction, Springer-Verlag.

Rousseeuw J P and Leroy M A, Eds. (1987). Robust Regression and Outlier Detection, John WIler & Sons.

Rueda S, Udupa J K, et al. (2010). "Shape modeling via local curvature scale." Pattern Recognition Letters(31): 324-336.

Scambos A.T, Dutkiewicz J. M, et al. (1992). "Application of Image Cross-Correlation to the Measurment of Glacier Velocity Using Satellite Image Data." Remote Sensing Of The Environment **42**: 177-186.

Schiewe J (2005). "Status and future perspectives of the application potential of digital airborne sensor systems." International Journal of Applied Earth Observation and Geoinformation **6**: 215-228.

Schuurman N, Grund D, et al. (2006). "Spatial/temporal mismatch: a conflation protocol for Canada Census spatial files." Canadian Geographer **50**(1): 74-84.

Skaloud J and Legat K (2008). "Theory and reality of direct georeferencing in national coordinates." ISPRS Journal of Photogrammetry & Remote Sensing **63**: 272-282.

Snow National Center Ice Data (2009). IMCORR, National Snow and Ice Data Center.

Super B J (2004). "Fast Correspondence-based system for shape-retrieval." Pattern Recognition Letters(25): 217-225.

TU Z, Zheng S, et al. (2008). "Shape matching and registration by data-driven EM." Computer Vision and Image Understanding(109): 290-304.

Vivek E P and Sudha N (2007). "Robust Hausdorff distance measure for face recognition." Pattern Recognition Letters(40): 431-442.

Weis M, Muller S, et al. (2005). "A framework for GIS and imagery data fusion in support of cartographic updating." Information Fusion **6**: 311-317.

Xiong D (2000). "A three-stage computational approach to network matching." Transportation Research part C(8): 71-89.

Xu M, Watanachaturaporn P, et al. (2005). "Decision tree regression for soft classification of remtore sensing data." Remote Sensing Of The Environment(97): 322-336.

Yuan X, Fu J, et al. (In Press). "The application of GPS precise point positioning technology in aerial triangulation." ISPRS Journal of Photogrammetry & Remote Sensing.

Zhang C (2004). "Towards an operational system for automated updating of road databased by intergration of imagery and geodata." ISPRS Journal of Photogrammetry & Remote Sensing(58): 166-186.

Zhang D and Lu G (2003). "A comparative study of curvature scale space and Fourier descriptors for shape-based image retrieval." Journal of Visual Communication and Image Representation **14**(1): 39-57.

Zhang J, Zhang X, et al. (2003). "Object representation and recognition in shape spaces." Pattern Recognition(36): 1143-1154.

Zitova B and Flusser J (2003). "Image registration methods: a survey." Image and Vision Computing **21**: 977-1000.